(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平6－295286

(43)公開日　平成６年(1994)10月21日

(51)Int.Cl.[5]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　15/16	３１０　Ｄ	7429－5L		
15/00	３１０　Ａ	7459－5L		
Ｈ０４Ｌ　9/00				
9/10				
		8949－5K	Ｈ０４Ｌ　9/00	Ｚ

審査請求　未請求　請求項の数47　ＯＬ　（全 43 頁）　最終頁に続く

(21)出願番号　特願平5－79302

(22)出願日　平成５年(1993)４月６日

(31)優先権主張番号　８６３５５２

(32)優先日　1992年４月６日

(33)優先権主張国　米国（ＵＳ）

(71)出願人　593067033
アディソン・エム・フィッシャー
ＡＤＤＩＳＯＮ　Ｍ．　ＦＩＳＣＨＥＲ
アメリカ合衆国、33942　フロリダ州、ナプルズ、フォーティーンス・アベニュ・サウス、60

(72)発明者　アディソン・エム・フィッシャー
アメリカ合衆国、33942　フロリダ州、ナプルズ、フォーティーンス・アベニュ・サウス、60

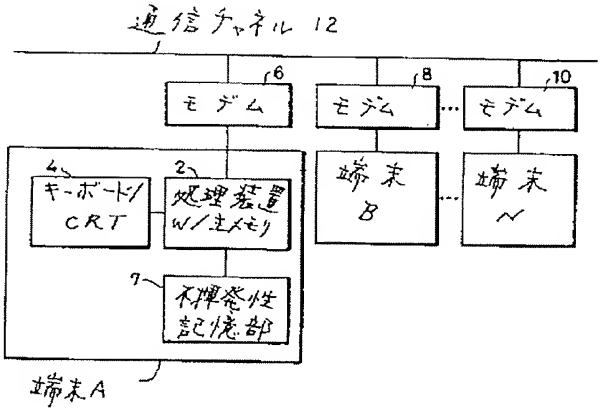
(74)代理人　弁理士　深見　久郎　（外３名）

(54)【発明の名称】　コミュニケーションシステムにおけるコンピュータ間で情報を処理するための方法

(57)【要約】

【目的】　移動プログラムを作成し、支援しかつ使用するための方法および装置を提供する。

【構成】　「移動プログラム」はディジタルデータ構造であり、それは一連の命令および関連するデータを含み、かつ少なくとも１つの次の行先または受信者を決定して移動プログラムを受信し、かつそのプログラムによって決定されたすべての関連あるデータとともにそれ自体を次の受信者または行先に送る能力を有する。移動プログラムは任意のアルゴリズムに従ってサインされるべきディジタル材料、および必要に応じて確証されるべきディジタル材料を計算することが可能である。



【特許請求の範囲】

【請求項１】　複数個のコンピュータ（端末Ａ、Ｂ、…、Ｎ）がそれを介してコンピュータがメッセージを交換可能なチャネル（１２）に結合されたコミュニケーションシステムにおいて、前記コンピュータ間で情報を処理するための方法であって、

第１のコンピュータに一連のプログラム命令（図２、ブロック２２）を与えるステップを含み、プログラム命令は第１のコンピュータによって実行され、組の命令を受信するべき少なくとも１つの次の行先を決定する命令を含み、前記組の命令は前記命令を添付のデータとともに前記次の行先に送信するための命令を含み、

ディジタル値を計算するステップを含み、その内容は前記プログラムによって実行される論理決定および処理に基づき、さらに少なくとも１つの行先で前記ディジタル値に基づいてディジタル署名（４３２）を実行するステップを含む、方法。

【請求項２】　前記ディジタル署名は前記一連のプログラム命令によって論理的に処理されることにさらされるデータとして表わされる、請求項１に記載の方法。

【請求項３】　前記ディジタル署名に関連するディジタル証明は前記一連のプログラム命令によって論理的に処理されることにさらされるデータとして表わされる、請求項１に記載の方法。

【請求項４】　前記ディジタル署名は次の行先に送信される前記添付データの一部として含まれる、請求項１に記載の方法。

【請求項５】　前記一連のプログラム命令によって前記データを認識された標準に少なくとも部分的に従う特殊化されたデータ構造に転換するステップをさらに含み、それによって前記データ構造は前記命令に関係なく有用である、請求項１に記載の方法。

【請求項６】　ディジタル署名およびそれが与えられるデータを、前記プログラム命令とは関係なく、処理しかつ確証するステップをさらに含む、請求項５に記載の方法。

【請求項７】　データは一連の命令の指示の下で標準化された電子データ交換（ＥＤＩ）フォーマットに転換される、請求項１に記載の方法。

【請求項８】　ディジタル署名が選択的に与えられ得る情報を論理的に組立てるステップを含み、かかる情報はそれに基づいて一連のプログラム命令が動作するプログラム変数として扱われる、請求項１に記載の方法。

【請求項９】　ディジタル署名は前記一連のプログラム命令の制御下で呼出され得る機能として実行される、請求項１に記載の方法。

【請求項１０】　前記ディジタル署名機能に供給されるデータは、ユーザのファイルから読出されたデータ、前記一連のプログラム命令にビルトインされたデータ、ユーザによって入力されたデータ、他の署名から得られたデータ、およびディジタル証明（５１４）から得られたデータのうちの任意の組に基づいて値を反映する、請求項９に記載の方法。

【請求項１１】　署名人によって行使された権限が適切に行使された（４３４）ことの確証を可能にするのに十分なディジタル情報を含むことによって、ディジタル署名（４３２）を実行するユーザに与えられた権限の表示をさらに含む、請求項１に記載の方法。

【請求項１２】　ディジタル証明の収集からディジタル署名（４３４）を実行する際に使用されるべき証明を選択するステップをさらに含む、請求項１に記載の方法。

【請求項１３】　複数個のコンピュータ（端末ａ、ｂ、…、Ｎ）がそれを介してコンピュータがメッセージを交換することが可能なチャネル（１２）に結合されたコミュニケーションシステムにおいて、前記コンピュータ間で情報を処理するための方法であって、

第１のコンピュータに一連の命令（図２、ブロック２２）を与えるステップを含み、命令は第１のコンピュータによって実行され、組の命令を受信するべき少なくとも１つの次の行先を決定する命令を含み、前記組の命令は添付のデータとともに前記命令を前記次の行先に送信するための命令を含み、

前記命令の実行によって前記コンピュータの少なくとも１つのユーザからデータを獲得するステップと、

前記データを認識された標準に少なくとも部分的に従う特殊化されたデータ構造に前記命令の実行を経て転換するステップとを含み、それによって前記データ構造は前記命令とは関係なく有用であり、さらに前記命令の実行を経て前記データ構造をディジタル的に署名するステップを含む、方法。

【請求項１４】　データは組の命令の指示の下で標準化された電子データ交換（ＥＤＩ）フォーマットに転換される、請求項１３に記載の方法。

【請求項１５】　ＥＤＩトランスレータを使用することによって転換するステップを含む、請求項１３に記載の方法。

【請求項１６】　ＥＤＩトランスレータは前記命令の制御下で呼出される外部モジュールである、請求項１５に記載の方法。

【請求項１７】　前記データ構造の集合体の少なくとも一部は前記データ構造のディジタル署名とともに組の命令とは関係なくデータの組として送信される、請求項１３に記載の方法。

【請求項１８】　前記データ構造の集合体の少なくとも一部は前記データ構造のディジタル署名とともに組の命令とは別個にストアされる、請求項１３に記載の方法。

【請求項１９】　ディジタル署名の結果は添付データの一部としてストアされる、請求項１３に記載の方法。

【請求項２０】　ディジタル署名の結果は組の命令が少なくとも１つの後続の行先で実行された場合に確証され

る、請求項１３に記載の方法。

【請求項２１】　複数個のコンピュータ（端末Ａ、Ｂ、…、Ｎ）がそれを介してコンピュータがメッセージを交換することが可能なチャネル（１２）に結合されるコミュニケーションシステムにおいて、前記コンピュータ間で情報を処理するための方法であって、

コンピュータに組の命令を受信するべき少なくとも１つの次の行先を決定する一連の命令を含む第１の移動プログラムを与えるステップを含み、前記組の命令は前記命令を添付データとともに前記次の行先に送信するための命令を含み、

少なくとも前記コンピュータのうちの１つに第２の移動プログラム（図３７：６９４）を与えるステップと、

第１の移動プログラムの指示の下で第２の移動プログラムを実行するステップとを含む、方法。

【請求項２２】　第１および第２の移動プログラムは同一の組の命令の異なったインスタンスである、請求項２１に記載の方法。

【請求項２３】　第１および第２の移動プログラムは全く違う組の命令を含む、請求項２１に記載の方法。

【請求項２４】　第１の移動プログラムは第２（図３７：６９４）によって実行されるべき動作を規定する第２の移動プログラムにデータを与える、請求項２１に記載の方法。

【請求項２５】　第２の移動プログラムはデータを第１の移動プログラム（図３７：６８７、７０６）に戻す、請求項２１に記載の方法。

【請求項２６】　移動プログラムおよびデータが様々なコンピュータシステムおよびハードウェアアーキテクチャに基づいて解釈され得るように、双方の移動プログラムがハイレベル解釈フォーマットで送信される、請求項２１に記載の方法。

【請求項２７】　解釈フォーマットは少なくとも２つの全く違う型のコンピュータで処理することが可能である、請求項２６に記載の方法。

【請求項２８】　第１の移動プログラムは第２の移動プログラムをメモリから消去する、請求項２１に記載の方法。

【請求項２９】　第２のプログラムインスタンスはその実行後保存される、請求項２１に記載の方法。

【請求項３０】　複数個のコンピュータ（端末Ａ、Ｂ、…、Ｎ）がそれを介してコンピュータがメッセージを交換することが可能なチャネル（１２）に結合されるコミュニケーションシステムにおいて、前記コンピュータ間で情報を処理するための方法であって、

コンピュータに一連の命令（図２、ブロック２２）を含む第１の移動プログラムインスタンスを与えるステップを含み、命令はコンピュータによって実行され、組の命令を受信するべき少なくとも１つの次の行先を決定する命令を含み、前記組の命令は前記命令を添付データとともに前記次の行先に送信するための命令を含み、

前記コンピュータの少なくとも１つに前記移動プログラムインスタンス（図３７：６９４）を与えるステップと、

第１の移動プログラムインスタンスの命令の指示の下で第２の移動プログラムを処理するステップとを含む、方法。

【請求項３１】　処理動作は第２の移動プログラムインスタンスを消去するステップを含む、請求項３０に記載の方法。

【請求項３２】　処理動作は第２の移動プログラムインスタンス（６８７、７０６）からデータを抽出するステップを含む、請求項３０に記載の方法。

【請求項３３】　処理動作は第２の移動プログラムインスタンスのプログラム命令を変更するステップを含む、請求項３０に記載の方法。

【請求項３４】　処理動作は第２の移動プログラムインスタンスにストアされた変数の値を変更するステップを含む、請求項３０に記載の方法。

【請求項３５】　前記第２のプログラムインスタンスは第１のプログラムインスタンスと同一の命令を含む、請求項３０に記載の方法。

【請求項３６】　複数個のコンピュータ（端末Ａ、Ｂ、…、Ｎ）がそれを介してコンピュータがメッセージを交換することが可能なチャネル（１２）に結合されるコミュニケーションシステムにおいて、前記コンピュータ間で情報を処理するための方法であって、

第１のコンピュータに一連の命令（図２、ブロック２２）を与えるステップを含み、命令はその第１のコンピュータによって実行され、組の命令を受信するべき少なくとも１つの次の行先を決定する命令を含み、前記組の命令は前記命令を添付データとともに前記次の行先に送信するための命令を含み、さらに前記一連の命令の実行に応答してファイルを選択するステップと、

前記選択されたデータファイルの内容の少なくとも一部を前記一連の命令（図３５、図３６、図３７、６４０－６４２、６８０－７０８、７１０－７３２）に応答して前記次の行先に送信するステップとを含む、方法。

【請求項３７】　前記ファイルのデータの少なくとも一部をディジタル的にサインするステップを含む、請求項３６に記載の方法。

【請求項３８】　前記ファイルのデータの少なくとも一部のハッシュ値を計算するステップを含む、請求項３６に記載の方法。

【請求項３９】　複数個のコンピュータ（端末Ａ、Ｂ、…、Ｎ）がそれを介してコンピュータがメッセージを交換することが可能なチャネル（１２）に結合されるコミュニケーションシステムにおいて、前記コミュニケーションシステムで情報を送るための方法であって、

第１のコンピュータに１組の命令（図２、ブロック２

２）を与えるステップを含み、その組の命令は前記組の命令の複数個のインスタンスを発生し、かつ前記インスタンスのうちの１つを添付データとともにそれぞれ受信する少なくとも第１および第２の行先への送信を開始する命令を含む第１のコンピュータによって実行され、さらに前記第１および第２の行先に送信された前記インスタンス内にその全く違う送信経路（図３７）の間に蓄積されたデータを後で併合する能力を含むステップを含む、方法。

【請求項４０】　１つのインスタンスをマスタインスタンス（６８２）として確立するステップと、さらにマスタを制御して他のインスタンスが併合行先に到着したときに他のインスタンスからデータを抽出するステップをさらに含む、請求項３９に記載の方法。

【請求項４１】　複数個のコンピュータ（端末Ａ、Ｂ、…、Ｎ）がそれを介してコンピュータがメッセージを交換することが可能なチャネル（１２）に結合されるコミュニケーションシステムにおいて、前記コンピュータ間で情報を処理するための方法であって、

第１のコンピュータに一連のプログラム命令（図２、ブロック２２）を与えるステップを含み、そのプログラム命令は第１のコンピュータによって実行され、その組の命令を受信するべき少なくとも１つの次の行先を決定する命令を含み、前記組の命令は前記命令を添付データとともに前記次の行先に送信するための命令を含み、さらに前記一連の命令が実行することを許容されるその組の動作を修飾するステップを含む、方法。

【請求項４２】　前記修飾手段は前記プログラムを使用する当事者によって指定される、請求項４１に記載の方法。

【請求項４３】　前記修飾手段は前記移動プログラムを使用する当事者によって信頼される１つの当事者によってディジタル的に署名される、請求項４１に記載の方法。

【請求項４４】　複数個のコンピュータ（端末Ａ、Ｂ、…、Ｎ）がそれを介してコンピュータがメッセージを交換することが可能なチャネル（１２）に結合されるコミュニケーションシステムにおいて、前記コンピュータ間で情報を処理するための方法であって、

第１のコンピュータに一連のプログラム命令（図２、ブロック２２）を与えるステップを含み、その一連のプログラム命令は第１のコンピュータによって実行され、その組の命令を受信するべき少なくとも１つの次の行先を決定する命令を含み、前記組の命令は前記命令を添付データとともに前記次の行先に送信するための命令を含み、さらにユーザトークンデバイスにストアされた個人のキーを使用することによってディジタル署名を実行するステップを含む、方法。

【請求項４５】　ユーザの個人のキーがトークンデバイスまたはコンピュータメモリにストアされているかどうかを決定するステップと、署名を前記トークンデバイスで、またはユーザのコンピュータでそれぞれ処理するステップとを含む、請求項４４に記載の方法。

【請求項４６】　複数個のコンピュータ（端末Ａ、Ｂ、…、Ｎ）がそれを介してコンピュータがメッセージを交換することが可能なチャネル（１２）に結合されるコミュニケーションシステムにおいて、前記コンピュータ間で情報を処理するための方法であって、

第１のコンピュータに一連のプログラム命令（図２、ブロック２２）を与えるステップを含み、その一連のプログラム命令は第１のコンピュータによって実行され、その組の命令を受信するべき少なくとも１つの次の行先を決定する命令を含み、前記組の命令は前記命令を添付データとともに前記次の行先に送信するための命令を含み、さらに日付／時間公証を実行するステップを含む、方法。

【請求項４７】　複数個のコンピュータ（端末ａ、ｂ、…、Ｎ）がそれを介してコンピュータがメッセージを交換することが可能なチャネル（１２）に結合されるコミュニケーションシステムにおいて、前記コンピュータ間で情報を処理するための方法であって、

第１のコンピュータに一連のプログラム命令（図１、２２）を与えるステップを含み、そのプログラム命令は第１のコンピュータによって実行され、その組の命令を受信するべき少なくとも１つの次の行先を決定する命令を含み、前記組の命令は前記命令を添付データとともに前記次の行先に送信するための命令を含み、さらに時間遅延機能（図９：５７０）を実行するステップを含む、方法。

【発明の詳細な説明】

【０００１】この発明は一人のコンピュータ使用者から他のコンピュータ使用者へ必要な関連データとともにそれ自体を移動させる能力を有し、それにより様々なコンピュータノードでのデータ処理、データ認証、およびデータ収集のための強力なツールとなる「移動」（travelling）プログラムを創造するための方法および装置に関する。

【０００２】

【発明の背景および要約】組織内においては、文書はしばしば人手を介して移動する。複数の組織間で文文書をやり取りする必要がある場合には、郵便または配達サービスがしばしば採用される。

【０００３】組織内および組織間での電子的文書伝送技術についてはよく知られている。電子郵便システム、電子伝達システム等の急速な発達により或る種の商取引が自動化されかつ多くの場合不必要である人手による文書の伝達の幾らかはなくなった。

【０００４】ユーザ間（たとえば組織内）で自動的に情報を伝達するための或る先行技術の方法論は、いわゆる「電子形式」方法論を利用する。この「電子形式」方法

論はユーザに対しデータを提供し、従来技術の表示装置によりユーザの入力を促し、入力されたデータが正確に入れられたことを確証しかつその後そのようなデータを他のユーザに伝送するというものである。

【０００５】多くの観点において、この電子形式の方法論は非常に限定されたものである。たとえば、データが何らかの値を表わしている場合、常にデータが操作もしくは改ざんまたは単に偽造される危険性がつきまとう。この危険性に対処する試みには、ディジタル的にサインされるべき或る種の重要なフィールドにフラグをつけることが含まれていた。これにより特定の入力フィールドについて或る限られた量ではあるが入れられたものと全く同じであるとする認証が可能となる。

【０００６】しかしながら、この方法では、複雑なデータ構造を組立てかつその後ディジタル的にサインすることは不可能である。本願は、いかなるアルゴリズム等に従っても、移動プログラムがサインされるべきディジタル資料および必要に応じて確証されるべきディジタル資料を計算することを可能にする。

【０００７】こうして、たとえば、本願発明によれば、サインされる実際のデータがいかなるフィールドデータ自体とも違うようになることが可能となる。実際、サインされた資料がユーザにより提供される実際のデータを全く含んでいないということも可能である。

【０００８】たとえば、特にこのことが有益である１つの態様は、本願発明の移動プログラムがエントリーされたデータの局面に基づきＥＤＩ（電子データ交換）トランザクションを創出する場合である。プログラムはＥＤＩトランザクションにサインする能力を有する。このようなＥＤＩトランザクションは他の表のファイルまたはこの移動プログラムを駆動するスーパバイザーまたはインタプリタからのプログラム内の内部表に基づきルックアップされた複雑なディジタル情報から構成され得る。こうして、或る種のテーブルから選択された「Ｘ」として単にエントリーされたかもしれない入力フィールドと、サインされる実際のディジタル資料は全く異なっている。

【０００９】上記のディジタル署名のタイプは寿命が長いデータ構成に対する応用が考えられ、かつ恐らくは或る期間を経て異なるものにより確証され得ると考えられる。たとえば、ＥＤＩの場合、署名はＥＤＩトランザクション自体に結びつけられることが可能で、かつこのトランザクションの将来の受け手により確証され、それは移動プログラムの脈絡を外れたものでもよい。このタイプのディジタル署名は紙の購入注文書または契約書の下の部分にある手書きによる署名に類似するものである。

【００１０】任意のデータにサインすることが可能な点に加えて、本願発明によれば、プログラムはどのユーザがこの署名プロセスに参加すべきかを何らかの知られた基準に基づき条件的に決定することを可能である。

【００１１】たとえば、本願発明によれば、この移動プログラムは特定のデータ、ユーザまたは組合せについてどのような共同署名の必要性が存在し得るかについてプログラムの範囲内で論理的な決定を下すことができる。これにはユーザのＸ．５００（certificate)証明または補強されたディジタル証明（たとえば本願発明人の米国特許第４，８６８，８７７号または第５，００５，２００号によるもの）に含まれる情報を含み得る。完全なプログラムとしてのフレキシビリティが存在するので、このように取り出された情報をこの移動プログラムの将来の伝送ルートの調整に使用することさえ可能である。

【００１２】簡単な認証のためにディジタル署名を使用する点に加えて、本願発明は、許可の要件および使用を含みかつ確証することをも可能にする。これはたとえば許可の証明および委任を管理するという第４，８６８，８７７号および第５，００５，２００号の教示を利用する。

【００１３】一方、本願発明はまたディジタル署名を使用して移動プログラムが他のタイプの貴重な認証を提供することを可能にする。たとえば、本願発明による保安の便宜として、一人のユーザから他のユーザへの伝送のすべてに関してディジタル署名認証を行なうことが可能となる。これは移動プログラム自体、その変数および補助的データまたはファイルを含む。

【００１４】この第２のタイプのディジタル認証は、部分的には、長期間の署名を保持するという点で上記のデータ中心の認証とは相違しており、というのも伝送される変数および他のデータが、一度受け手側のユーザが行動をとった場合に変更されるからである。この第２のタイプの認証はしたがって基本的には不正の防止策として考えられかつまたこの形式の実際のユーザの一人によるものであっても許可されない不正を法的に過去に遡って監査するために使用され得る。

【００１５】加えて、本願発明はまた、いかなるバグまたは「ウィルス」も侵入していないことを確認するために、何らかの信頼のおける発行権者（たとえば著者）により移動プログラム自体にサイン、認証および許可が行なわれ得る第３のタイプの認証を提供する（これはルートに沿ったプログラムの有効な所有権を有する使用者による感染さえも防止する）。

【００１６】本願発明は、使用者のグループの間でのデータ収集の自動化のための独自の機構を提供する。移動プログラムが一人のユーザに送られ、関連するデータファイルを取り付け（または切り離し）かつ次のユーザへと移動し得る。一人または二人以上のユーザから収集されたデータまたはファイルは他のユーザにより保管されるかまたは所望であればバッチ処理のために蓄積され得る。この方法論により個々のユーザがすべての必要なデータを必要なフォーマットで伝送することに頼る必要がなくなる。

【００１７】この発明はまた組織内でのユーザから次のユーザへそれ自体が送られる移動プログラムの脈絡における電子文書交換（ＥＤＩ）と、データの収集、編集、および是認を効果的に行なう。プログラムの論理により決定される適当な時点で、他の組織へ伝送するために標準ＥＤＩトランザクション（たとえばＸ１２　８５０購入注文トランザクションセット）をプログラムで発生することが可能である。移動プログラムは終了したトランザクションのセットにディジタル的にサインすることが可能である。したがって、標準化されたＥＤＩおよび標準化された署名を処理し得る受け手側の組織であれば、その受け手側の組織が本願発明により教示される入手可能なすべての強力な技術を有していなくても、入来する資料を認証しかつ処理することが可能となる。

【００１８】逆に、この発明により、一般的なＥＤＩトランザクション、恐らくはサインされたものを移動プログラムが受けかつそれらが構文解析されその変数に組込まれることを可能にする。移動プログラムはそこで入力を確証し、表示装置内に組込み、かつ必要に応じて様々な受け手の間を移動させる能力を有する。

【００１９】本願発明のこれらのおよび他の特徴は添付の図面とともに見るとき、以下の発明の好ましい実施例の説明を読むことによりより良く理解されることであろう。

【００２０】

【好ましい実施例の詳細な説明】図１は本願発明に関し使用可能な例示的通信システムを示すブロック図である。このシステムは端末Ａ、Ｂ、…、Ｎがその上で通信を行ない得る通信チャネル１２を含む。通信チャネル１２は、たとえば電話線等の保安処理がなされていない通信チャネルでもよい。

【００２１】端末Ａ、Ｂ、…、Ｎは例示目的のみにより従来技術のキーボード／ＣＲＴ表示装置４に結合されるプロセッサ（メインメモリ）を有するＩＢＭ　ＰＣ互換性コンピュータでもよい。メインメモリ２を有するプロセッサはディスクメモリ等の不揮発性記憶装置にも結合される。各端末Ａ、Ｂ、…、Ｎはまた従来技術のモデム（それぞれ６、８、１０）に結合されると、移動プログラムを含むメッセージを端子が伝送および受信することを可能にする従来技術のＩＢＭ　ＰＣ通信ボード（図示せず）を含む。

【００２２】本明細書中で使用する、「移動プログラム」はディジタルデータ構造であって、連続する命令と関連データとを含み、かつ移動プログラムを受けかつそれ自体をプログラムにより決定されるすべての関連データとともに次の受け手または行先へ伝送するための少なくとも１つの次の行先または受け手を決定する能力を有する。

【００２３】各端末はメッセージを発生し、その移動プログラムに固有の論理、データ、および機能をロードしかつ実行するのに必要ないかなるディジタル署名動作をも達成し（これについては本明細書内でより詳細に説明する）、かつそのメッセージを通信チャネル１２（または通信チャネル１２に接続され得る通信ネットワーク（図示せず））に接続された他の端末に伝送することができる。

【００２４】発明者の米国特許第４，８６８，８７７号および５，００５，２００号ならびに第５，００１，７５２号に記載されるディジタル署名および証明方法論がここで使用されてもよく、これら特許についてはここに引用により援用する。代替的には、より従来技術のディジタル署名方法論が使用されてもよい。

【００２５】本願発明の例示的実施例に従う「移動プログラム」構造および方法論をより詳細に論じる前に、実際の商取引の脈絡における一般的な動作の例について簡単に説明する。まず、図１の端末Ａのユーザが、回路設計プロジェクトを完成するためにコンポーネント部品を入手しようとしている会社における設計チームの一員である比較的地位の低い技術者と想定する。

【００２６】キーボード４を使用するこの技術者が以下に詳細に説明するようなタイプの部品要求「移動プログラム」をアクセスしたとする。要求「移動プログラム」は技術者に対し必要とされるコンポーネント部品を説明するように促す。移動プログラムはそれ自体を自動的に次の行先、たとえば端末Ｂへのアクセスを有し、この組織構造内ではより高い地位にあり、この要求を認めかつディジタル的にこれにサインする権限を有している上役に自動的に伝送する命令シーケンスを含む。移動プログラムはまた補助的な情報たとえば将来の行先で必要なまたは役に立つかもしれないファイルを伝送してもよい。上役はこの要求に適切にディジタル的にサインを行なうよう促される。ディジタル署名が単に特定の可変の値だけではなく可変の名称をも反映することは可能である。代替的には、プログラム内で計算される変数から生じる何らかの集合的構造を反映してもよく、その値がファイルから読出されたデータ、ユーザによる入力、プログラム内に内蔵されたデータ、様々な署名者の証明、またはユーザの環境から取出されるデータ（たとえばユーザのＩＤ）等を含む多くの源のいずれかに基づいていてもよい。

【００２７】要求が認められれば、要求の形式は組織内ではこれが認められない場合とは違った経路をとることになる。移動プログラムは組織内でそれ自体をどこへ伝送すべきかを、動作端末Ｂでの上役からの入力に基づいて決定する知能を有し得る。移動プログラムはまた所望であれば要求に関連する適切なデータを、端末Ｂに関連するメモリにロードしかつ所望であれば組織内のどこか先に進める必要がある端末Ｂからのいずれかのファイルを取付けることになる。

【００２８】一度署名がなされると、移動プログラムは

その後いつでも、その後のどのユーザに対しても、かついずれの理由であっても、資料が確証されるように再計算し、かつディジタル署名確証を行なう能力を有する。

【００２９】このような確証の結果はいずれの受け手に対しても告知されることが可能で、または多くの場合には、移動プログラムは単に確証を行ないかつ（データの改ざんが行なわれたことを示唆する）不良があれば、問題を告知する。

【００３０】移動プログラムモニタは第４，８６８，８７７号および第５，００５，２００号の教示を実施し得るので、いずれの受け手も必要な許可が行なわれたことを確認することができるように、許可についてもチェックすることが可能である。

【００３１】特定のデータ構造が構築されかつ移動プログラムの制御の下サインが行なわれた後、引続きこのデータ構造を再構築しかつその署名を他のものに与えることが可能である。このようなデータはこの後はいずれのものによっても改ざんされ得ない。

【００３２】しかしながら、この発明はまたすべての伝送されたデータが一人のユーザから次のユーザに送られる際にディジタル的にサインされるという能力を実現する。受け手のコンピュータ内の移動プログラム処理装置は移動プログラムがロードされると自動的にこの署名を確証することができる。これによりコンポーネント等が途中で変更されたり改ざんされたりすることはないことを確実にする。この全体的署名のみがこの特定の伝送の間のデータの状態を表わし、かつその後のユーザにとってはいかなる意味ももたないが、これは第三者に邪魔されない完璧な伝送を確保するとともに、この形式を所有している参加ユーザによる改ざんを追跡する必要がある場合には法的監査機構を提供する。この全体的署名はこの署名が移動プログラム自体によって伝送プロセスの一部として条件的に誘引され得るという点において電子郵便がサインされる現在の能力とは異なるものである。

【００３３】最終的には、組織構造内ですべての許可が得られた後、移動プロセスは実際の購入注文をつくり出すことになる。

【００３４】これは多くの態様でなされ得る。移動プログラムが幾つかの方法を支持し、所与の状況に応じた１つの最も適切なものを選択してもよい。ここでは４つの可能性について述べる。

【００３５】１．移動プログラムは物理的に郵送されると考えられる用紙の上に最終購入注文を単にプリントアウトし得る（会社のロゴ、レターヘッド等をプリントすることさえも可能）。

【００３６】２．移動プログラムは、外へ向けてのコンピューターファックス能力に結合されると、販売者のファックス装置上に現われると考えられる購入注文画像を自動的に発生することが可能である。購入者は用紙を生成する必要がない。

【００３７】３．販売者がまた本願発明の移動プログラム方法論を支持することがわかっていれば、移動プログラムが販売者を単に次の行先に指定することも可能である。

【００３８】４．販売者が本願発明を使用せずまたは購入者の移動プログラムが、販売者がこの移動プログラムの方法論を取り扱う能力があるかどうかはっきり判断できないという可能性も高い。

【００３９】したがって、移動プログラムはその内部データを操作して標準ＥＤＩ（電子データ交換）トランザクションを構築し、これが広い範囲で認識かつ処理される。移動プログラムはまたコンピュータＥＤＩトランザクションに対しディジタル署名をさせてもよく、かつ署名およびトランザクションの双方が伝送され得る。移動プログラムはその後ＥＤＩトランザクションと何らかの可能な署名とを受け手に対し伝送する（このような伝送はその方向づけられた移動の一部としてのユーザからユーザへの移動プログラムおよびその付属するものの伝送とは独立しており、かつこれと混同されるべきではない）。

【００４０】標準ＥＤＩトランザクションを取り扱うことができる受け手ならば誰でも受け取ったＥＤＩ入力を扱うことができる。ディジタル署名を取り扱うことが可能な受け手であれば誰でもこのトランザクションを認証することができる。さらに、受け手がこれらを認識するに足るソフトウェア能力を有している場合には、この受け手は署名の一部として実施され得る認証を自動的に確証することができる。どの範囲まで証明がサインされたトランザクションとともに伝送されるかは移動プログラムの論理による。

【００４１】上記のいずれの場合においても、幾つかの可能なレベルの自動化のいずれかを使用して、移動プログラムは販売者への購入注文（Ｐ．Ｏ．）情報をスピンオフすることが可能である。これに引続いて、移動プログラムはそれ自体の１つのバージョンまたは恐らくは単なる文字を発生者に戻し、Ｐ．Ｏ．が送られたことを告知する。他の情報はアーカイブか、またはさらなる処理を待つために待合せに送られ得る。この情報は簡単なメッセージ、ファイルに加えられた記録で、さもなくば恐らくは移動プログラムがフルのトラバーサル（自動「メイリング」または「伝送」）を予定する。

【００４２】図２は本願発明の例示的実施例に従う移動プログラムとその関連するコンポーネントの構造を示す。図２の移動プログラムは少なくとも以下の多重フィールドのセグメントを含む。第１のヘッダセグメント２０は好ましくはコンポーネントセグメントの各々の大きさ、関連するプログラム（かつできれば以下に述べる他のセグメント）の名称、日付、各コンポーネントのタイプ（たとえばプログラムはソース言語プログラムかまたはプログラムは既にコンパイルされたＰコードか等）、

形式の識別、それを行なうために必要なインタプリタのバージョン、プログラム再開の適切なポイントで実行を再開することが必要なデータ（たとえば実行スタック、ＰＣＢ等）、最も最近のトラバーサルに関連する日付、およびプログラム許可情報（ＰＡＩ）を識別する。移動プログラム構造内の各セグメントは「各コンポーネントのタイプおよびサイズ」フィールド「Ｓ」がヘッダセグメント２０内に含まれることがないようにそれ自体の表現を含み得る。本願の目的に従い、プログラム許可情報（ＰＡＩ）は関連するプログラムが行なうことが許可される動作の範囲を規定する保安情報とみなされ得る（たとえばファイルへのアクセスの定義、プログラム呼出能力、電子郵便発生能力、データを他のユーザへ発送する能力、文書をリリースする能力、機械言語プログラムを実行する能力、メモリの特定の領域をアクセスする能力、ユーザに対し情報を表示する能力、ディジタル署名を要求する能力、ディジタル公証装置（digital notary public)等にアクセスする能力である）。プログラム許可情報の性質および使用に関してのさらなる詳細については本願出願人の出願である「プログラム許可情報を利用するコンピュータシステム法および装置」という名称の出願（代理人事件番号２６４－２９）に見られる。ヘッダセグメント２０はまた関連の移動プログラムのバージョン番号を含む。

【００４３】移動プログラム構造２２セグメントは実施例のヘッダに従いかつ好ましくは再構築された外部の実施プログラム言語（たとえばＲＥＸＸ言語）またはＰＡＳＣＡＬもしくはＣＯＢＯＬに近いもので書かれる。プログラム自体はたとえば購入注文に関連する応用に関したものでもよい。

【００４４】移動プログラムはそれ自体をさらなる受け手に送る能力を含む上記に述べたような特徴を有することになる。したがって、プログラム２２は一人または二人以上の受け手に入手可能な何らかの媒体を経由してそれ自体を先に送る命令を含むことになり、これが本明細書中では「トラバーサル」として知られる。１つのソースコード命令または幾つかのＰコード命令が一人または二人以上の識別された受け手に対して移動プログラムを結果として「トラバーサル」するのに必要かもしれない。図２に示される移動プログラム構造はいかなる特定のコンピュータアーキテクチャからも独立するように設計されておりかつ国際標準（たとえばＸ．２０９フォーマット）に従い構成されている。

【００４５】移動プログラムはまた「変数セグメント」２４を含む。第１のユーザにより実行される前に、変数セグメント２４はほとんど空でもよい。一度プログラムが受け手に送られると、さらなる変数がプログラムにより要求されるとおりに規定されることになりそれによりプログラムがさらに実行されると変数の数が増えることになる。例示目的のみで、変数セグメント２４は或る変数たとえば「トータル．ドル．受領」をこの変数の実際のデータ値とともに識別し得る。

【００４６】各変数は図２に示されるフィールド３２－４２の各々に示される関連情報を有し得る。フィールド３２は変数名称の大きさを識別する。変数名称自体はフィールド３４に記憶される。変数の値の大きさはフィールド３６に示される。変数の値はフィールド３８内にある。フィールド４０は変数が属する実行スタックレベルを識別する。実行スタックレベルが識別されるのは、同じ変数名称がプログラム内の異なるレベルに存在し得るからである（たとえば１つの変数名称は第１のサブルーチン内に存在してもよく、かつ同じ変数名称が別のまたは集まったサブルーチン内に存在しかつしかも異なる定義を有し得る）。実行スタックレベルは受け手のコンピュータ内の移動プログラムを再構築して送り手のコンピュータ内で有していたものと同じ論理構造をとらせるために有益である。フィールド４２は変数のタイプ、たとえばストリング、８ビットバイト、整数等を識別し得る随意のフィールドである。

【００４７】「変数」セグメント２４はまたそれぞれの変数および関連情報のディジタル署名を含み得る。こうして、１つまたは２つ以上の変数が移動プログラムの実行経路の間の様々な時点でとられたディジタル署名を反映することが可能となる。本願発明の重要な局面の１つは、移動プログラムがどのようなタイプの情報に対してもディジタル署名をつくり出すことが可能な点である。この署名自体が変数として運ばれる。この署名を確証するためには、プログラムがサインされた正確な値を示し（またはできれば再計算し）、その後それを署名値（これも変数で表わされる）とともに通過させて移動プログラムの確証署名（VERIFYSIGNATURE)機能に送ることが必要となる。変数のディジタル署名を含むことにより、受け手側は、データが１）不正を受けておらず、２）有効なサインを受け、かつ３）サインした人物が適切に許可を受けたものであったことを確証することが可能となる。許可をディジタル署名と関連づけるための好ましい機構を記載する上記の米国特許第５，００５，２００号を参照されたい。

【００４８】たとえば、上記の米国特許第５，００５，２００号に記載のとおり受け手によりいかなる署名でも確証され得るように、図２のセグメント２６は移動プログラムに何らかのディジタル署名と関連する証明を含んで示される。代替的には、証明はディジタル署名とともに「変数」セクションに含まれ得る。

【００４９】セグメント２８Ａ－２８Ｎは、移動プログラムが移動プログラムのユーザに属するファイルを取り付けかつ記憶することが可能になるように記録されかつ名前をつけられたファイル画像を含む。その後、ユーザのファイルは移動プログラムとともに他の先行するユーザのファイルとともに伝送され得る。ファイルに名前を

つけることでユーザによるファイルの後々のアクセスが容易となりかつ移動プログラムのユーザがたとえばさらに伝送されるファイルまたは特定のユーザが特定の状況下で保管するファイルを識別することが可能となる。

【００５０】移動プログラムはまた「変数セグメント」３０を含み、これはたとえば受け手が、移動構造全体の伝送が最後に送られてからは不正を受けていないことを確証し得るように移動構造全体のディジタル署名を含む。移動プログラムデータ構造について記載してきたが、ここで移動プログラムの実行の間に使用されるデータ構造と移動プログラムを実行するための関連ソフトウェアについて説明する。実行制御領域（ＸＣＡ）データ構造が図３に示される。ＸＣＡは一度移動プログラムが受け手により受け取られかつＰコードにコンパイルされると（最初からＰコードで送られた場合は除く）移動プログラムを実行するプログラムにより要求される情報を特定する。

【００５１】図３に示されるとおり、ＸＣＡセグメント８２は入来のファイル内に現われたプログラムのアドレスおよび大きさを識別する。本明細書全体を通して、或るセグメントが「アドレス」または「位置」を記憶するものとして記述される場合には、データは物理的または論理的アドレスでかつ実際の物理的メモリ位置を必ずしも直接的に特定する必要がない点を認識されたい。プログラムはソースまたはＰコードで受け取られかつどちらであるかについての表示が維持される。実行制御領域はプログラムのＰコード版のアドレスとその大きさを表わすセグメント８４を含む。現在のプログラム制御ブロックのアドレス（またはアドレスへのポインタ）はセグメント８６内で識別される。たとえば使用されるファイル制御ブロック（ＦＣＢ）のリストを移動プログラムに関連するファイルに取り付けかつ取り外す位置はセグメント８８に示される。移動プログラムに取り付けられる証明を制御するために使用される証明制御領域（ＣＣＡ）のアドレスはセグメント９０内に示される。「変数」情報表（ＶＩＴ）の位置は変数を「Ｂ－ツリー」の形式で制御かつ維持するセグメント９２内に示され、このツリーは各プログラム「変数」の位置を識別する階層２進ツリー構造である。

【００５２】実行制御領域はまた移動プログラム内に存在する認証と許可とを確証するために使用され得る保安情報セグメント９４を含む。セグメント９６はアクセスを必要とするかもしれない入来の移動プログラムを含むファイルの名称を規定する。セグメント９８はプログラム自体が入来経路に沿ってメイルされた回数を追い続ける。実行制御領域はまた入力パラメータセクション１１０を含み、これによりプログラムの実行に関連するパラメータが識別され得る。実行制御領域セグメント１０２はヘッダ情報が入手可能になるように移動プログラムファイルから受け取った入力ベッダ情報を識別する。

【００５３】図４は移動プログラムがファイルをそれ自体に取り付けまたは取り外すときに使用されるファイル制御ブロック（ＦＣＢ）のデータ構造を示す。ファイル制御ブロックは特定のユーザのシステム内で取り付けられるかまたは外される特定のファイルを指すタグを識別するタグフィールド１１６を含む。ファイル制御ブロックもまた次のファイル制御ブロックへのポインタであるセグメント１１０を含む。ファイル制御ブロックはまた関連するファイルが受け取られた移動プログラムによりちょうど取り付けられたものかどうか、ファイルが次のトラバーサルの際（たとえば次のメイリング）の際に取り外され得るか否か、ファイルが外へ出たか（すなわち関連のファイル画像が別個のユーザファイル内にロードされたか）どうか等の様々な状態を規定する状態セグメント１１２と、ファイルが流れ中心（stream oriented）のものかまたは記録中心（record oriented)のものか等の「ファイルのタイプ」に関するインジケータとを有する。ファイルの他の属性はこのフィールド内で規定され得る。

【００５４】セグメント１１４は問題の特定のファイルがすばやくアクセスされ得るように主の入来の移動プログラムファイル内のファイルの位置に関する表示を記憶する。セグメント１１８はファイルのローカル名（すなわち移動プログラムの最も最近の受け手により識別されたファイル名）を識別する。ファイルのローカル名は、ファイルが取り付けられかつさらなる受け手に送られるかまたは既に取り付けられたファイルが「外に出されている」、すなわち特定のユーザにより局所的に記憶されている場合に典型的に与えられるものである。加えて、図４に示されるとおり、ＦＣＢは関連ファイルのハッシュを含んでもよい。当業者により理解されるとおり、ハッシュとは基礎となるデータを与えられれば、コンピュータでは計算が容易なはずである「１方向」機能である。ハッシュ値を与えられればハッシュ機能は計算的には基礎となるデータを決定することもそのハッシュとして特定される値を有するいかなるデータをつくり出すことも不可能なはずである。すべての実用目的に関して、元のデータの集合体にハッシュ機能を与えることから得られた値は元のデータの偽造不可能な独自のフィンガプリントである。元のデータがいずれかの態様で変更された場合には、その修正されたデータのハッシュは異なるものになる。

【００５５】図５は移動プログラムの実行の間に使用され得る例示的プログラム制御ブロックを示す。プログラム制御ブロックは１つのルーチンが他のルーチンを呼出し、各ルーチンが関連のプログラム制御ブロックを有する構成化されたプログラミングコンテキストにおいて実行されているプログラムに関する制御情報の追跡を続ける。

【００５６】プログラム制御ブロックセグメント５０は

プログラム実行制御スタックにおける先行するプログラム制御ブロックを指し示す。プログラム制御ブロックは現在の実行プログラム内において実行されるべき次のＰコード位置を規定するセグメント５２を含みセグメント５４は行なわれた最後のＰコード動作のタイプを規定する。セグメント５６は表現評価の間に使用される表現評価スタックに対するポインタを含む。実行スタックは、実行スタックが表現の評価と内部状態の追跡のために使用されるという点でプログラムスタックとは典型的に異なっている。セグメント５８はこのスタックプログラムのレベルを規定しかつセグメント６０は共有される変数のリストに対するポインタを規定する。ＲＥＸＸ言語においては、「露出された」ステートメントが共有された変数をアクセスするために使用され得る。

【００５７】図６は変数を制御するために使用される可変制御ブロックデータ構造（ＶＣＢ）を示す。セグメント６２はＢツリーのうちどこに変数が位置しかつ幾つかのポインタを含み得るかを識別する。セグメント６４は変数値の大きさを識別しかつセグメント６６はその値がメモリのどこに位置するかを示すポインタを識別する。セグメント６８は変数のタイプを識別するために随意に使用され得る。セグメント７０は移動プログラムのどのレベルに変数が関連しているかを識別して、プログラムが実行された後に、プログラムに関連していたいずれかの位置を示す変数が容易に削除され得るようにする。セグメント７６および８０は変数の名称の大きさおよび名称それぞれを識別する。

【００５８】ここで、移動プログラムの実行の説明に戻る。インタプリタ実行駆動プログラムの「ローダ」部により行なわれる動作のシーケンスが図７－１２に示される。これらの動作は移動プログラムを実行するための準備に関連する。

【００５９】移動プログラムは対話型ユーザモード、すなわち他のプログラムにより呼出されるモード、または情報を収集しながらノードからノードへ送られるバッチ動作モード等の複数の異なるモードの１つにおいて実行し得る。初期化情報は特定の動作モードおよび関連する実行時間パラメータを識別するスタートアップ動作（１２０）の間に入力される。

【００６０】図７－１２に示されるフローチャートは図２に示される移動プログラム構造がどのようにしてロードされるかを示す。移動プログラムをロードする際に、インタプリタが実行制御領域ＸＣＡと初期プログラム制御ブロックＰＣＢとをつくり出す。これにより入力パラメータへのアクセスが省かれ、変数情報表（ＶＩＴ）をロードしかつ初期化するために与えられていた入力ファイルの名称が省かれる（１２２）。フローチャートブロック１２２において、実行制御領域および移動プログラムに関連するプログラム制御ブロックが設定される。様々なＸＣＡおよびＰＣＢフィールドが引続く処理の間に満たされる。

【００６１】この後、ローダは、図２に示されるような移動プログラムのセグメント、すなわちヘッダ、プログラム、変数、証明、ファイルおよびクロージャセグメントのローディングを開始する。たとえばヘッダプログラム等の上記の移動プログラムセグメントの各々をロードすることにより以下のように適切なデータが充填されることになる。

【００６２】ブロック１２４では、より多くのセグメントを処理する必要があるかどうかの判断が行なわれる。もし必要であれば、初期入力はそのセグメントに関して読出されかつセグメントのタイプが判定され、その後セグメントの処理がセグメントのタイプに応じて開始される（１２６）。

【００６３】図８のヘッダ処理に戻ると、処理されているセグメントが第１のセグメントかどうかを判断するためのチェックが行なわれる（１５０）。もしそうでない場合には、ヘッダが最初のセグメントでなければならないので、エラー状態が存在することになる（１５２）。第１のセグメントが処理されている場合には、ヘッダが読出されハッシュされる。ヘッダのデータがＸＣＡ内に記憶される（１５４）。

【００６４】ルーチンはここでエントリーポイントＬで分岐して図７に戻る。ローダは処理されるべきセグメントがまだあるかどうかを判定する（１２４）。もしそうならばブロック１２６が実行されて図９に示すような「プログラム」セグメントの処理が結果として行なわれる。最初、ヘッダが存在するかかつまだプログラムはロードされていないかどうかを判定するチェックが行なわれる（１６０）。もし答えがノーであれば、エラー状態が存在することになる（１６２）。もし答えがイエスの場合には、プログラムが読出されかつハッシュがとられる（１６４）。

【００６５】その後、プログラムハッシュおよび／またはプログラムに関連するディジタル署名（および／またはヘッダ）が確証される（１６６）。ディジタル署名が適正に許可されておらずまたは確証されない場合には、エラー状態が生じる（１６６）。確証が発生すると、移動プログラムに関する保安および許可情報が蓄えられる（１７０）。代替的には、このような許可情報はヘッダまたはプログラムセグメント内に保持され得る。

【００６６】ブロック１７２においてプログラムがＰコードとして送られたかどうかを判定するチェックが行なわれる。Ｐコードではなくソースコードが送られた場合には、ソースコードは当業者に周知の従来技術のコンパイラ技術を利用してＰコードにコンパイルされかつソースコード画像は記憶１７４から削除される。ブロック１７２でのチェックにかかわらず、プログラムの入来ファイルにおける位置――ソースであるかＰコードフォーマットであるかにかかわらず――がＸＣＡ内に蓄えられ

る。入来画像の位置および範囲を知ることで最終的な外へ向かうトラバーサルにプログラムをコピーすることが簡略化される。最終的には、Ｐコードが単にコンパイルされたか否か、または入来ファイルから読出されたか否かにかかわらず、Ｐコードの主記憶アドレスおよび大きさが１７８の実行制御領域（ＸＣＡ）内に設定され、その後図７に示されるルーチンがブロック１２４に再びエントリーされて、これにより図１０に示す「変数」セグメント処理等の残ったセグメントのローディングが生じる。

【００６７】ブロック１９０に示されるような「変数」セグメントの処理において、ヘッダおよびプログラムはロードされているが先行する変数は何もないかどうかを決定するチェックが行なわれる。もしそうでない場合、エラー条件は結果的に１９２になる。もしヘッダおよびプログラムはロードされているが先行する変数が何もなければ、もしあればすべての変数を読出す反復処理が開始される。１９４で読出すべき変数が（さらに）あるか否かを決定するチェックが行なわれる。もし読出すべき変数がさらにあれば、各変数について変数制御ブロック（ＶＣＢ）が図６に示されるように作成され、変数制御ブロック（ＶＣＢ）への変数識別子および値の挿入ならびにＶＣＢにおける或る状態条件のセットによって完成される。さらに、変数制御ブロックは変数情報テーブル（ＶＩＴ）の適切なスポットに加えられ、そのテーブルはすべてのプログラム変数を含む（１９６）。

【００６８】さらに、たとえば移動プログラムの前の実行に関する他の変数情報が１９８において適切にメモリスタックまたはプログラム制御ブロック内にロードされる。代替的に、そのような「制御」情報をここよりもヘッダセグメントに保持することが望ましいかもしれない。その後、ルーチンはブロック１９４に分岐して戻り、そこでさらなる変数を読出すことが必要であるか否かを決定するチェックが行なわれる。この処理は変数がもはや読出される必要がなくなるまで継続し、必要がなくなった点でルーチンは図７のブロック１２４に分岐して戻り、その結果次のセグメントをロードする。

【００６９】図１１に示されるように、各証明が読出され（２００）、証明エレメントが作成され、それは記憶装置の証明制御エリア（ＣＣＡ）に加えられる（２０２）。図１１に概略的に示されるように、すべての証明が受け取られるまでこのプロセスは反復され、その時点でルーチンはブロック１２４に分岐して戻りさらなるセグメントの検査を行なう。

【００７０】代替的に、プログラムセグメントより前に証明セグメントを伝送し、それによってプログラム認証／許可の一部として使用される証明がプログラム変数およびユーザからユーザへの認証によって使用されるあらゆる証明とともに維持され得ることが望ましいであろう。

【００７１】このブランチ動作の結果、図１２に示される「ファイル」セグメント処理が行なわれる。ファイルセグメントは典型的に「変数」セグメントに追従するので、変数セグメントが（たとえ空であっても）既にロードされたか否かを決定するチェックが行なわれる。もしそうでなければ、エラーが検出されており、適切なエラーメッセージが２１２で発生される。もし「変数」セグメントが既にロードされていれば、ブロック２１４に示されるように、そのファイルに関連のファイルタグが既にロードされたか否かを決定するチェックが行なわれる。もしそうであれば、２１６でエラーが検出され、ファイルが二重にされていることを示す。

【００７２】もしファイルタグが既にロードされていなければ、ブロック２１８に示されるように、ファイル制御ブロックがファイルのために組立てられ、タグ名がセットされ、移動プログラムと既に関連しているかもしれない他の状態識別子がセットされ、ファイル位置が入ってくるファイルに関してセットされる。

【００７３】その後、ファイルが読出され、そのハッシュが計算され、ＦＣＢのセグメント１１５に保管される。ファイルのサイズはＦＣＢのセグメント１１４に保管される。ファイルはこのときメモリ内にロードされる必要がない（２２０）。その後、作成されているファイル制御ブロックがＸＣＡに収集されたファイル制御ブロックリストに加えられ、このルーチンはブロック１２４に分岐して戻り次のセグメント（恐らく「クロージャ」）を処理する。

【００７４】図１３の「クロージャ」処理において、前のセグメントの各々についての前のすべてのハッシュからハッシュが計算される（２３０）。すべての「セグメント」材料がハッシュを施されて読出されることが認識されるべきである。ブロック２３２において、２３０でとられ、計算されたハッシュが、（クロージャセグメントに記憶されている）移動プログラムが送られたときに加えられたハッシュと一致しているか否かを決定するチェックが行なわれる。もし一致していなければ、エラー条件は２３４になる。

【００７５】もし一致していれば、移動プログラムがサインされた否かに関するチェックが行なわれる（２３６）。もしそうでなければ、ブロック２３８に示されるように、伝送データが全くサインされないという通知をあるいは提示するような、所望されるいかなるレベルの機密保護をも組込む処置が行なわれる（２３８）。

【００７６】もし伝送がサインされたならば、２４０で署名が確証され、移動プログラム（および関連の購買注文または他の形式）を実際に送った当事者を正確に識別するメッセージがユーザに提示される。このルーチンは次の図７のブロック１２４に分岐して戻る。

【００７７】図１３の「クロージャ」処理が完了すると、ブロック１２４で処理されるべきセグメントがもう

ないかが決定される。その後、クロージャがうまく受信され、処理されたか否かを決定するチェックが行なわれる（１２８）。もしそうでなければ、このルーチンは不首尾な有効性のチェックを行ない（１３０）、１３２で停止を処理した後、実行を中止する。

【００７８】もしブロック１２８におけるチェックでクロージャがうまく完了したことが明らかになると、プログラム実行の準備のための様々なステップが行なわれる（１３４）。この点で、スタックが復元され、変数情報テーブルおよび変数制御ブロックが復元される。プログラム制御ブロックは実行再開点を含むように復元される。

【００７９】その後、図１４に示されるルーチンが開始され、Ｐコード命令を実際に処理する。次の問題がここで検討されねばならない。すなわちプログラム実行が送り手の機械から（トラバーサルの一部として）伝送されたときと同じ状態で有効に復元されるため、移動プログラムが送る機械内にあり、送る機械自体から戻ったところであるか、または受け手の機械で復元されたところであるかをいかにして区別し得るかという問題である。

【００８０】この発明は多数の方法でこの問題を処理することを許容する。もしトラバーサル機能がビルトイン機能として実現されれば、インタプリタは特別な値（たとえば「０」）をプログラムへ、それがそれ自身をうまく送った後に戻し、別の値（たとえば「１」）をプログラムへ、その実行が受け手の機械上で復元されるときに戻すであろう。移動プログラムはこの状況を区別するためにこの値をテストすることができる。この区別が行なわれ得る別の方法は、移動プログラムに「先行トラバーサルの数」を抽出する機能を与えることによる（ＸＣＡのセグメント９８）。このトラバーサルを行なう前に、プログラムはこの機能を使用して先行ートラバーサルー計数機能を保管することができる。もしそれが変数の値と一致すれば、プログラムは送り手のコンピュータで実行が再開していることを知り、もしそれが異なっていれば（１だけ大きいはずであるが）、プログラムは受け手のコンピュータで実行が再開していることがわかる。

【００８１】第１のユーザが移動プログラムを発生するとき、図７－１３に示されるローダルーチンは極めて少ない変数、ファイル、または証明で、あるいはそれらなしで実行される。したがって、前述のステップのあるものは最初の処理の間省略されるであろう。ローダルーチンは移動プログラムが初めて実行されるか、さらなる受け手によって実行されるかにかかわらず実行される。

【００８２】図１４はＰコード命令の処理において行なわれる動作を示す。これは実行されるあらゆるＰコード命令について反復される。ブロック２５０に示されるように、次のＰコード命令のロケーションが現在のＰＣＢ（５２）から引出され、これが「現在の」Ｐコード動作になる。ブロック２５２において、Ｐコード動作の長さが決定され、「次のＰコード」位置（５２）が更新され、後に続くＰコード命令を反映する。この型の現在のＰコード動作は（５４）に保管される（これはインタプリタが正確な動作に基づいて僅かな変化しか有さない共通ルーチンを共用するのに有用である。たとえば「コール」動作および「機能呼出」動作が、機能呼出がパラメータが戻されることを期待する以外は同じである。）。

【００８３】その後、ブロック２５４に示されるように、示されるＰコード動作が行なわれる。ほとんどのＰコード機能はデータ操作、論理テストおよびプログラムフロー制御を含む。例示のみによって、このようなＰコード動作は変数を位置決めして、スタックにその変数をプッシュすることと、次のＰコード動作をリセットしてそれによってブランチ動作において生じるであろうようなフロー制御を変化させることと、演算またはストリング動作を行なうことと、ポップされたスタック値に基づいてＩＦ／ＴＨＥＮ／ＥＬＳＥ動作を行なうことと、スタック値に基づいてＤＯ／ＩＴＥＲＡＴＥ／ＵＮＴＩＬ／ＷＨＩＬＥ、または他の動作を行なうことと、スタック値に基づいてＳＥＬＥＣＴ／ＷＨＥＮ／ＯＴＨＥＲＷＩＳＥ動作を行なうことと、「ＥＮＤ」動作を行なってＤＯ／ＷＨＥＮ／ＳＥＬＥＣＴ動作を閉じることとを含んでもよい。

【００８４】この発明の独特の動作に関連して様々なＰコード動作が詳細に説明される。ここに与えられた手引によって、Ｐコード機能はインタプリタの書込の精通者によって簡単な態様で実現され得る。

【００８５】しかし、特定のＰコード機能の詳細をしばらく無視すると、好ましい設計はＰコード動作がそれらの完了時に論理「割込」を発生することを許容する。

【００８６】これらはＰコード処理の処理が何か他の外部動作が行なわれねばならない間、中断されることを許容する。この割込概念は好ましい設計において冗長な待機、または外部作業が呼出されるときは常に作業記憶のロールアウトを容易にするために使用される。

【００８７】図１５において、ブロック２５６でＰコードルーチンから戻ると、インタプリタはルーチンが論理割込の信号を送ったか否かを決定する。もし送っていなければ、２５０に戻り、次のＰコード動作を取扱う。

【００８８】もし割込が示されたならば、ブロック２５８の特別なチェックが、これが特別な「ＥＸＩＴ」要求であるか否かを決定するために行なわれる。もしそうであれば、記憶、ファイル、変数、ロードサブルーチンなどのようなこのプログラムの終わりで解放されるべきすべての資源がブロック２６０で廃棄される。２６０によって保管されているであろうＰコード動作からの可能性のある戻り値がブロック２５９でこの移動プログラムの呼出人へ戻される。

【００８９】これがＥＸＩＴではないと仮定すると、ブロック２６１はＲＯＬＬＯＵＴが行なわれるべきか否か

を決定する。たとえば或る環境において、作業記憶には、ユーザが入力を入れるのを完了する間、または移動プログラムが時間間隔が満期になるのを待機している間、または冗長な（もしくは大きい）外部プログラムが移動プログラム論理から呼出されている間、またはディジタル署名ルーチンが実行されている間（それがユーザ入力を含むことが多いので）、ロールアウトされることが有用である。

【００９０】Ｐコード割込および可能性のあるＲＯＬＬＯＵＴを生じるルーチンは、それらがビルトイン機能として、または言語ステートメント（それら自身のＰコードを含む）として組込まれるか否かにかかわらず、次のものを含む。

【００９１】ＳＩＧＮ
いかなるコンピュータデータへもディジタル署名を与え、そうすることにおいて、ユーザに多数の証明から選択することを請求し、かつユーザに個人の署名が解読され、使用されることを許容するユーザの秘密のパスワードキーを与えることを請求するであろう。

【００９２】ＤＩＳＰＬＡＹ
画面を構成、かつ出力し、ユーザの入力供給を待機する。

【００９３】ＴＩＭＥＷＡＩＴ
未来時間に到達するまで実行を中断する。
ＳＥＬＥＣＴ．ＦＲＯＭ．ＤＩＲＥＣＴＯＲＹ
たとえばユーザのディレクトリ、またはファイルのディレクトリなどからの選択を許容する。
ＮＯＴＡＲＩＺＥ
時間公認装置がそれ自身のディジタル署名を与えるのを待機する。

【００９４】或る環境において、ＲＯＬＬＯＵＴは不適切であり、これらの場合、ブロック２６２、２６４、２６８におけるロールアウトおよびロールイン処理は存在しないか、または禁止されるであろう。

【００９５】いかなる場合も、「割込」の信号を送るＰコード動作はまた少なくとも３つの関連の（「コールバック」）機能のアドレスを供給する。

【００９６】－－　プレーロールアウトルーチン。ロールアウトの準備において必要ないかなる機能も行なう。これは一時記憶にパームフィールドを準備して・・・へ送ることを含んでもよい。

【００９７】－－　インターーロールアウトルーチン。できる限り多くの作業記憶が補助記憶へロールアウトされた後実行する。

【００９８】－－　ポストー待機ルーチン。インターーロールアウトルーチンが終了した後、および作業記憶が補助から復元された後ロールバックの後の詳細を取扱う。典型的にはこれは、一時記憶に残っており、かつ実行上にロードされるか、またはプログラム変数内にコピーされねばならないインターーロールアウトルーチンにより計算された結果値をコピーすることを含む。

【００９９】ブロック２６１において、プレーロールアウトルーチンが呼出される。これは空のルーチンであるか、またはたとえばインターーロールアウトルーチンのためのパラメータをセットアップしてもよい。

【０１００】ブロック２６２において、もし環境および状況が適当に与えられれば、ロールアウト機能が行なわれる。もし行なわれれば、ＲＯＬＬＯＵＴはＶＣＢおよびそれらの値、ＦＣＢ、証明およびＣＣＡ、実行スタック、ＶＩＴ、ＸＣＡ、Ｐコード自体、ならびに他のいかなるブロックをも含むすべての作業記憶を（ファイルのような）ある補助記憶へ書込むことからなる。インタプリタ自体は記憶から開放されてもよく、十分な残余プログラムおよびデータがインタプリタおよび作業記憶を後に再ロードするために残っていると仮定すると、これは特別なブロック（２６４）で行なわれてもよい。

【０１０１】ステップ２６６において、インターーロールアウトルーチンが呼出される。典型的にはこのルーチンはユーザが入力を行なうこと、または未来時間もしくは他の事象まで待機すること、または入力を待機する、もしくは他の遅延を引き起こす、もしくはＲＯＬＬＯＵＴによって空にされた大きい記憶を要求するであろう別のプログラムを呼出すことを待機する。

【０１０２】ブロック２６８において、インターーロールアウトが終了した後、インタプリタが再ロードされ、実行スタック、すべての制御ブロック、Ｐコードを含む作業記憶が補助記憶から復元される。

【０１０３】次にブロック２７０において、いかなる最終処理も行なわれ、動作を一掃する。たとえば、これは典型的にインターーロールアウトルーチンにより戻された結果を実行スタックへ、またはプログラム「変数」へコピーすることを含む。

【０１０４】これは割込を完了し、次に制御がＰコードハンドラ（２５０）の先頭へ戻され、そこで次のＰコード命令が処理される。

【０１０５】これより関係のある幾つかのＰコード動作が検証される。好ましい実施例のインタプリタは３つのＣＡＬＬおよび機能を取扱う。すなわちインタプリタへ「ビルトイン」されたルーチン、移動プログラムの一部として書込まれるルーチン、およびインタプリタまたはプログラムの外部にあり、プログラムが実行されるとき動的に位置決めされ、呼出されるルーチンである。

【０１０６】図１７において、ビルトイン機能がかなり簡単に表わされ、インタプリタはＰコードのインデックスに基づき特定の機能を単に位置決めし、（インタプリタ内の）ルーチンのアドレスを検索し、それを呼出す。しかし、ほとんどのものは行なわないが、幾つかのビルトイン機能がＰコード割込の信号を送るかもしれないということを認識することが重要である。この場合、ビルトイン機能は必要なプレーロールアウト、インターーロ

ールアウトおよびポストー待機ルーチンを与えねばならない。

【０１０７】Ｐコードインタプリタは常にＣＡＬＬおよび機能を区別し、機能の場合においてのみ実行スタックへ結果を戻すことに備える。たとえば、ＳＩＧＮ機能は供給されたデータ上で計算されたディジタル署名を表わす値を戻す。

【０１０８】図１６（Ａ）においてプログラムルーチンへの呼出／機能によって新しいＰＣＢ実行レベルが３００で作成される。新しいＰＣＢはルーチンのＰコードエントリ点へ次のＰコード命令（５２）をセットすることによって、サブルーチンの開始時に実行を開始するようにセットされる。ルーチンの最初の命令はブロック２５０に再び入るときアクセスされるであろう。パラメータがプログラムルーチンのために準備され、適当な状態条件がセットされ、ＰＣＢのプログラムレベル５８が呼出プログラムより１つ高くセットされ、ＰＣＢは今現在のＰＣＢ（８２）として実行スタックの先頭に置かれる。プログラムルーチンの結果はＰコードＲＥＴＵＲＮ動作を通して呼出人へ送られる。

【０１０９】図１６（Ｂ）において、対応するプログラムＲＥＴＵＲＮ　Ｐコード動作がいかに動作するかが見られる。ブロック１２００はＲＥＴＵＲＮが最高の（唯一の）レベルのＰＣＢからつくられるかを決定し、その場合これはＥＸＩＴとして動作し、ブロック１２０４はＰコード「ＥＸＩＴ」割込が要求されているという信号を送り、プログラム全体のＲＥＳＵＬＴとしてブロック２６１によって最終的に戻される（図１５）べき値として（もしあれば）戻りＲＥＳＵＬＴを送る。

【０１１０】さもなければ、ブロック１２０４において、（たとえば呼出人のＰＣＢでフィールド５４をチェックすることによって）呼出人がＣＡＬＬまたは機能を使用したか否かに関する決定が行なわれ、後者の場合ブロック１２０６がスタック上に戻りＶＡＬＵＥを置く（またはＲＥＴＵＲＮがオペランドを有さなければ省略時の値を作成する）。

【０１１１】ブロック１２０８において、現在のレベルが一掃され、記憶域、ファイル、変数などを含むこのサブルーチンに個人的な（別名「プログラムレベル」）すべての資源が開放される。呼出人と共用されない変数のような資源は開放されず、利用可能である。

【０１１２】ブロック１２１０において、現在のＰＣＢが開放され、それによって呼出人のＰＣＢが現在のものとなり、ブロック２５６へ戻り、そこで実行が再開する。

【０１１３】インタプリタはビルトインルーチンを含み、これらは移動プログラムへディジタル署名、ユーザファイルを与えることに関する特定の移動プログラム関連機能、および他の機能を達成して、移動プログラムの設計者がこのような機能をプログラミングすることに関与する必要性を除去する。

【０１１４】Ｐコード動作はプログラム制御に影響を及ぼすＲＥＴＵＲＮ機能の性能およびプログラム制御ブロックに関するＰＲＯＣ動作を含んでもよい。インタプリタはこの中に説明される対話式表示方法論／言語を使用するＤＩＳＰＬＡＹ動作も行なう。インタプリタはＴＲＡＶＥＲＳＥ動作も行ない、その結果すべての関連データと同様別の受け手へ移動プログラムが「メイリング」される。

【０１１５】図１８は外部機能または呼出を実行するために行なわれる動作のシーケンスを例示的に示す。このような外部機能または呼出はインタプリタ、または移動プログラムの一部へは組込まれないが、むしろユーザのプログラムライブラリの一部に組込まれる。名前をつけられた機能または呼出は幾つかの可能性のあるライブラリのいずれかから３５４で位置決めされる。

【０１１６】３５６においてプログラムが見つかったか決定するチェックが行なわれる。もしプログラムが見つからなければ、所望ならばプログラムが終了したか、または省略時の処置が行なわれたかを決定するチェックが３５８で行なわれてもよい。もし終了の決定がなされると、エラーメッセージが発生され、様々なハウスキーピング／クリーンアップ動作が前述のように行なわれた後プログラムが出される（３６０、３６２）。

【０１１７】もしブロック３５８のチェックが省略時の処置がとられるべきであると示せば、たとえば特別な省略時機能値を戻す（３６８）ことによって省略時の処置がとられ、ルーチンは図１４のノード０へ分岐して戻り、さらなるＰコード命令の実行を開始する。

【０１１８】もしプログラムがブロック３５６で行なわれたチェックの結果見つかれば、パラメータがプログラムによって構成される（３６４）。外部ルーチンの呼出は可能性のあるロールアウトとともにＰコード割込を含む。これは、もし外部プログラムが冗長であるか、またはいかなる環境においてももし外部ルーチンが膨大であると、複数のユーザのスワッピング環境において記憶を保存することを許容し、したがって移動プログラムによって使用される記憶は外部プログラムを満足に行なうために空にされるべきである。この場合、Ｐコード割込がブロック３６６で信号によって送られる。示されるＰＲＥ－ＲＯＬＬＯＵＴルーチンはスタック（または変数）から一時記憶へ外部へコピーされる。ＩＮＴＥＲ－ＲＯＬＬＯＵＴルーチンはＥＸＴＥＲＮＡＬルーチンを呼出し、いかなる戻された結果をも受け、ＰＯＳＴ－ＷＡＩＴルーチンは（もし外部ルーチンが機能として呼出されたならば）戻された結果をスタックへコピーする。

【０１１９】外部ルーチンは実際に別の移動プログラムであることが可能である。もしそうであれば、Ｐコードインタプリタの既存の既にロードされた画像を使用し、新しいパラメータの組をブロック１２０へ単に送ること

によって特別な最適化が行なわれてもよい（図７）。この場合、特別な論理がブロック２６２および２６４に挿入され、インタプリタのコード自体を開放することを条件付きで回避することが必要であろう。

【０１２０】これよりこの実施例によって使用される様々な特別のビルトイン機能に注意が向けられる。これらの多くはビルトイン機能として、またはそれら自身の特別なＰコード動作を伴う言語ステートメントとしてのいずれかで実行され得る。

【０１２１】図２０および２１は移動プログラムがそれ自身を予め定められた受け手へ伝送するとき行なわれる動作を示す。ブロック３９８において、いかなるプログラム許可情報もまずチェックされ、トラバーサル動作が許容されることを保証する。（僅かな移動プログラムは移動することを許容されないかもしれないが、単に最初の使用で終了する幾つかの機能を行なうことは許容されることが考えられる。）プログラムが移動することを許容されない稀な場合には、特別な戻りコードが呼出人へ与えられる。

【０１２２】この実施例はビルトイン機能として「ＴＲＡＶＥＲＳＥ」動作を実現する。さらに、この機能は機能の直接の呼出人へ「０」を戻し、機能が受け手のコンピュータで再開された後呼出人へ「１」を戻すように規定される。先に説明したように、戻りコードのこの違いはプログラムが送り手および受け手のコンピュータ間の区別をすることを許容する。

【０１２３】これを行なうためにブロック３９９では、ＴＲＡＶＥＲＳＥ機能がまず実行スタック上に値「１」を予めロードし、スタックがそのままで伝送されることがわかる。これはしたがって移動プログラムが受け手のコンピュータ上で再構成され、再開されたとき戻されるであろう値である。「変数」情報テーブル、処理制御ブロック、様々なスタック、変数制御ブロックのようなすべての関連の変数データが、図２に示されるフォーマットのような伝送フォーマット内に集められる。

【０１２４】ブロック４０２で示されるように、移動プログラムヘッダが構成され、伝送される。移動プログラムはセグメントごとに伝送されるが、実際にはフィールドごとのフォーマットで、または所望ならば何か別の方法で伝送され得る。好ましくは、ハッシュが各セグメントについてそれが伝送されるに伴いとられる。

【０１２５】その後、４０４においてプログラムおよび移動プログラムとともに受取られた入力ファイルからのいかなる許可情報も出力伝送ファイルへコピーされる。「変数」セグメントは各変数の名前、現在値、および関連の状態を含んで伝送される（４０６）。この、または前のトラバーサル中ディジタル（許可）署名を行なう一部として集められたいかなる証明も伝送される。それゆえ、ディジタル署名動作が行なわれるときはいつも、４０８においてすべての関連の証明が移動プログラムの証明セクションに収集され、伝送される。署名はプログラム内で（すなわち変数制御ブロック内で）変数として保持される。現在好ましい実施例における証明はビルトイン機能呼出を介してアクセスされ得る材料として扱われる。

【０１２６】代替的に、プログラム自体を認証し、許可する署名および全体の伝送の署名に関する証明でさえも証明パッケージ内に含むことが可能であろう。しかし、これはすべての証明が証明セグメントが書込まれたときに明瞭にわかることが必要であり、セグメントの論理および恐らく位置が最適な処理を保証するために再び順序付けされることが必要であろう。

【０１２７】この実現において、ヘッダまたはプログラムセグメント内のプログラム許可情報によりプログラムの許可署名に関する証明と、クロージャセグメント内の署名によりユーザからユーザへの伝送署名許可のための証明とを保持することが好ましい。

【０１２８】証明が伝送された後、すべてのファイル制御ブロックが調べられ、その結果４１０において前のトラバーサル中に伝送されているであろうすべてのファイルおよび新たに取付けられたファイルのいかなるものの検査も行なわれる。ブロック４１２において調べるべきファイル制御ブロックがまだあるか否かを決定するチェックが行なわれる。ブロック４１４で調べられたファイルがスケジュールされて切り離されたか否かを決定するチェックが行なわれる。もしそうであればルーチンは４１２へ分岐して戻り、ファイルまたはファイルタグのいずれも伝送のためにコピーされない。もしファイルがスケジュールされ、切り離されなければ、４１６でファイルタグ名が伝送内にコピーされる。

【０１２９】関係のファイルが前方に運ばれている入ってくる移動プログラムの一部であるか否かを決定するチェックが行なわれる（４１８）。もしそれが入ってくるトラバーサルの一部であったと決定されれば、ファイル自体と同様入ってくるトラバーサルからのすべてのファイル属性が外部への伝送ファイルにコピーされる（４２２）。４２２でこの入力ファイル名は実行制御エリアＸＣＡを介してアクセスされてもよく、ファイルの入力位置はファイル制御ブロックに関連する。

【０１３０】もしファイルが入ってくるトラバーサルの一部ではないが、むしろ移動プログラム実行中に取付けられたものならば、ファイル、ファイルの型、およびその属性が４２０において伝送ファイル内にコピーされる。その後、ルーチンはブロック４１２に分岐して戻り、すべてのファイル制御ブロックが調べられるまで、調べるべきファイル制御ブロックがまだあるか否かを決定する。

【０１３１】図２１に示されるように、すべてのＦＣＢのものが調べられると、全体のユーザからユーザへのディジタル署名がシステムプログラムによって要求されて

いるかまたは必要とされているか否かを決定するチェックが４３０で行なわれる。このような全体の署名は伝送された情報との変更を検出することにおいて有用であろう。もし全体のディジタル署名がとられれば、伝送されたすべての材料のハッシュ上のディジタル署名動作が行なわれる（４３２）。ディジタル署名動作は米国特許第5,005,200号（または所望されるような関連の許可認証属性を有さない慣用的なディジタル署名技術）の教示に従って行なわれてもよい。ブロック４３２に示されるように、ハッシュは予め伝送の各部分についてとられた。代替的にハッシュがハッシュの各々についてとられてもよいことが注目される。ディジタル署名ステップは署名を行なうユーザ対話を含んでもよい。

【０１３２】その後、正当性の立証が「クロージャ」セグメントとして伝送の終わりに与えられる。正当性の立証は前の材料を反映するハッシュを伝送することによって与えられる。サインされたハッシュは４３４においてユーザからユーザへの認証を示すべきである。最終署名の正当性を立証するのに必要ないかなる証明も証明セグメント内には既になく、ＣＬＯＳＵＲＥセグメントに含まれるべきである。その後、伝送が４３６で閉じられる。

【０１３３】最後にブロック４３７で、伝送されたプログラムのためにそれが受け手のところに到着したとき実行スタック上に予めロードされた値「１」が除去され、現在の呼出人へ戻されてそれがそれ自身を区別することを許容する値「０」と置換えられる。

【０１３４】ディジタル署名を作成することは典型的に、恐らく証明を選択して個人のキーを開けるか、またはユーザに彼のディジタル署名トークン装置を作動することを求めることのようなユーザ対話を含み、図２０および２１に説明される材料が実際に好ましい実施例においてＰコード割込ルーチンとして作動するであろう。一例として、ＴＲＡＶＥＲＳＥ機能コードがＰコード割込をトリガし、そこでブロック３９９から４３０までの論理がＰＲＥ－ＲＯＬＬＯＵＴルーチンとして作動するが、４３２のブロックは、前述のユーザ対話を必要とするであろうためＩＮＴＥＲ－ＲＯＬＬＯＵＴルーチンとして作動するかもしれない。ブロックはその後（４３４など）ＰＯＳＴ－ＷＡＩＴルーチンとして作動する。

【０１３５】移動プログラムはそれ自身を何度もその実行中に様々な受け手へ伝送することが所望されるように設計され得る。このような多重伝送において、変数は各伝送に先立ち適切に変更され得る。この態様において、処理を行なう位置にあるプログラムは実現依存の態様で各受け手について異なる。

【０１３６】図２２は移動プログラムへファイルを取付けるための動作のシーケンスを示す。取付ファイルルーチンは識別されたファイルタグおよび識別されたファイル名に応答する。ブロック４４０で示されるように、同じタグを有するファイル制御ブロックが存在するか否かを決定するチェックが行なわれる。もしそうであれば、同じタグを有する前のファイルが４４２で削除される。

【０１３７】その後、特定のファイル名がユーザによってアクセス可能な既存のファイルを反映するか否かを決定するチェックが行なわれる。この点で、移動プログラムはファイルにアクセスする能力を含むプログラムが行なうことができる動作範囲を規定するプログラム許可情報に関するものであってもよい。

【０１３８】このようなプログラム許可情報はファイル名がアクセス可能であるか否かを決定するためにチェックされるであろう。もしファイル名がユーザによってアクセス可能でなければ、エラーコード／メッセージが４４６でユーザへ戻される。

【０１３９】もしファイル名がユーザにアクセス可能であれば、ファイル制御ブロック（ＦＣＢ）が特定のタグおよびファイル名によって組立てられ、ファイルが４４８で次の、かつ後に続く移動プログラムの伝送中に取付けられるであろう。ルーチンはその後ファイルがうまく取付けられていることを示して再開される。

【０１４０】図２３はユーザシステムからファイルがいかにして消去されるかを示す。「消去」機能が実行されようとするとき、機密保護コードが、プログラムがこのような動作を行なうことを許可されたかどうかを決定するためにチェックされる（４５０）。もし機密保護コードが、プログラムが特定のファイルを消去することを許可されたことを示せば（４５２）、消去動作が行なわれ、４５４でルーチンはファイルがうまく消去されたか否かを示し、分岐して戻る。代替的に、もしプログラムが消去動作を行なうことを許可されなければ、呼出ルーチンが、ファイルが消去されることができないということを示すエラーメッセージとともに戻される（４５６）。

【０１４１】図２４はファイルを移動プログラムから切り離す際行なわれる動作のシーケンスを示す。ブロック４５８に示されるように、切り離されるべきファイルに関する識別されたタグについてファイル制御ブロックが存在するか否かを決定するチェックが行なわれる。もしＦＣＢが存在しなければ、主ルーチンが、ファイルが切り離されることができないということを示すエラーメッセージとともに４６２で戻される。もしファイル制御ブロックが４５８で決定されるように存在すれば、ファイル制御ブロックは４６０で削除され、主ルーチンが、ファイルがうまく切り離されたということを示して戻される。

【０１４２】図２５は、ファイルが「移出」されるべきとき、すなわちユーザファイルへ変形されるとき行なわれる動作のシーケンスを示す。移動プログラムはたとえば簡易言語を表わす特定のファイルをとり、移動プログラムがさらなる行先へ送られた後もこのようなファイル

をユーザとともに残る受け手ユーザのファイルへ変換してもよい。「移出」されるべきファイルはタグおよび出力ファイル名、ならびにもし所望であればファイルが再び書込まれてもよいか否かを識別する再書込インジケータによって識別されるであろう。

【０１４３】４９８でまずファイル制御ブロックが特定のタグについて存在するか否かがチェックされる。もしＦＣＢが存在しなければ、適切なエラー表示コードが発生され、呼出ルーチンが（５０４）へ戻る。もしＦＣＢが特定のタグとともに存在しなければ、ファイルが入ってくる移動プログラムの一部であるか否かを決定するチェックが５００で行なわれる。もし移出されるべきファイルが入ってくるトラバーサルの一部でなかったならば、それはユーザによって取付けられており、既にユーザのファイル内に存在しているにちがいなく、したがって５０２で新たに取付けられたファイルを移出することが許容されないということを示すエラーメッセージが発生される。もしファイルが入ってくるトラバーサルの一部であったならば、特定のファイルが既に存在しているか否かを決定するチェックが行なわれる（４８０）。もしそうであれば、ブロック４８２で特定のファイルを再び書込むことが許可されるか否かを決定するチェックが行なわれる。このチェックはプログラムが（もし「オーバライティング」がなければ）特定の既存ファイルを変更するか、または（もし「オーバライティング」が許容されれば）特定のファイルを消去し、作成するかいずれが許容されるかを決定することを含む。もしそうでなければブロック４８４はプログラムへアクセスエラーを戻すために使用される。もし４８２のチェックが再書込を許可すれば、ファイルがオーバライティングされるべきか、または新たな材料がファイルの終わりに加えられるべきかに関する決定が行なわれる（４８６）。もしオーバライティングが４８６で示されれば、既存ファイルが消去される（４８８）。もしプログラム許可機密保全情報によって許可されれば新たなファイルが作成され、ファイルの初めに書込を開始するように準備がなされる（４９０）。

【０１４４】もしオーバライティングが４８６で示されないが、新たな材料データが終わりに加えられるべきであれば、既存のファイルの終わりに加えることを開始する準備がブロック４９２に示されるように行なわれる。その後、データは入ってくるトラバーサルファイルにおける正しい位置から出力ファイルへコピーされ（４９４）、移出動作がうまく行なわれていることを示し、主ルーチンが再び入れられる（４９６）。

【０１４５】図２６は材料がディジタル的にサインされるべきとき行なわれる動作のシーケンスが例示的に示される。ディジタル署名機能の実現において、まずディジタルサイン動作がブロック５１０で示されるようにプログラムによって許可されるか否かを決定するチェックが行なわれる。プログラムがディジタル署名動作を行なうことを許可されるか否かは、プログラムに関連し、かつプログラムが許可を与えられない動作が行なわれないことを確実にするように実行するとき常に監視されるプログラム許可情報によって制御される。もしディジタル署名動作が許可されなければ、５１１でディジタル署名機能を拒絶するエラーメッセージが発生される。

【０１４６】もしディジタル署名動作が許可されれば、ブロック５１４においてＳＩＧＮ機能が、実際の署名を行なうことに関連するユーザ対話を行なうであろうＩＮＴＥＲ－ＲＯＬＬＯＵＴルーチン（図２７に示される）による受信の準備において一時記憶へ（データ内容に必要とされる許可のような）パラメータとともにサインされるべきデータの画像を移動することによってユーザ対話の準備をする。

【０１４７】ブロック５１２においてＰコードルーチンは後述の割込ルーチンによって信号を送られる。

【０１４８】もしディジタル署名許可が許可されれば、ユーザへどの証明が署名動作のために使用されるべきであるかを請求するために表示パネルが提示されねばならない。署名動作は引用によってここに明確に援用されている本発明者の米国特許第5,005,200号に従って好ましくは行なわれる。ユーザは米国特許第5,005,200号の線に沿って構成されたものを含むディジタル署名動作を行なうために広範囲の証明を所有してもよい。ＩＮＴＥＲ－ＲＯＬＬＯＵＴルーチンはブロック５０９で記憶の多くがロールアウトされた後（署名ルーチン自体がむろん記憶内に残らねばならない）、制御が与えられる。

【０１４９】もし署名を行なうために適当な証明がなければ、制御はブロック５１５に進み、サイン動作へ戻されるべきエラーインジケータを発生する。もし署名を行なうのに適当な証明が１つだけあれば、それは自動的に（５１３）へ送られる。もし１つより多くの適当な証明があれば、ユーザは選択することが求められる（５１６）。もしユーザが断れば（５１７）、適切なエラーインジケータが発生され、プログラムへ送られる（５１５）。さもなければ選択された適当な証明が（５１３）へ送られる。

【０１５０】関連の個人のキーが次に位置決めされる（５１３）。もしブロック５１８がユーザのトークン上にそれが位置決めされることを決定すれば、それがディジタル署名を行なうことができるようにトークンへ通信を請求するためにステップ（５２４）が使用される。そうでなければ、ユーザの個人のキーは秘密のパスワードフレーズ下で暗号化されたシステムにおいて位置決めされる。ユーザはこのパスワードを請求され（５２０）、これは個人のキーを解読するために使用される。エラーまたは無効なパスワードが検出されれば、適切なエラーメッセージが発生される。本物のユーザ以外の誰かによって推量することを禁じるために、正しいパスワードを

与える試行は限られた回数しか許容されない。

【０１５１】ブロック５２２においてパスワードは個人のキーを解読するために使用され、これはひいては必要な許可に従ってメッセージをサインするために使用される。この動作の後、秘密の材料のすべての痕跡が消去され、署名および証明が一時記憶に戻される（２６８、図１５）。（２７０）において一時記憶から実行スタックへ署名を動かすＰＯＳＴ－ＷＡＩＴルーチン（５３０）へ制御が与えられる。

【０１５２】ブロック５３２において、動作が検査され、もしうまく行なわれていれば、サインする人の証明のプルーフ階層が得られる。証明はもし既に現われていなければ、（ＸＣＡ（９０など）に保持される）全体の証明収集物へ加えられる。

【０１５３】図２８は情報をユーザへ表示する際行なわれる動作のシーケンスを示す。移動プログラムは図２８に関連して説明される表示レイアウト能力とともに関連される。移動プログラムのレイアウト能力は付加的に高められた能力とともにユーザの対話式表示モードにおける使用のためのアプリケーションをタイプセットすることに従来関連していた機能を適用する。

【０１５４】画面は入力フィールドが容易に移動され、ユーザとの極めて融通性のある対話を行なうための様々な属性と関連され得るようにレイアウトされてもよい。様々な表示に関連する動作および機能はブロック５４０に要約される。表示はインタプリタの表示処理部分によって制御される特定のレイアウト定義処理に基づき出力を提示する。

【０１５５】表示処理はレイアウト定義におけるフィールドおよびフィールドの集まりの条件属性および静的属性の分析を含む。表示処理サブルーチンにおいて、条件論理を使用する変数置換および反復が必要に応じて行なわれる。変数置換が許可されているが、このシステムは、たとえフィールドがレイアウト定義によって指示されるようにその最終出力位置へ流れ込まされても、入力変数とそのフィールドが対応する変数制御ブロック（ＶＣＢ）における画面上の表示されるべき場所との間の結合を保持する。

【０１５６】色、フォント、ボールド体（boldface）／イタリック、様式、サイズ、下線、ブリンク、反転映像、非表示（たとえばパスワードの隠蔽のため）、高輝度表示などを含む属性が各フィールドへ与えられる。さらに、可能性のあるエラーメッセージが検出されたエラー条件に適当な場所に挿入され、適切なカーソル位置が示される。

【０１５７】ブロック５４０に使用されるレイアウト言語は画面出力の定義だけではなく入力を受け取るための定義も許可する。ブロック５４２に示されるように、アプリケーションによって適切に、入力フィールドを許容するユーザの端末へフィールドが書込まれる。前に述べたように、データ構造は、ユーザが適切な入力フィールドへのデータエントリーを行なった後に補助記憶へロールアウトされ（５４４）、ロールバック（５４６）されてもよい。

【０１５８】このために、ステップ５４４はＰコード割込の信号を実際に送り、ブロック５４５を関連のＩＮＴＥＲ－ＲＯＬＬＯＵＴルーチンとして、かつブロック５４６を関連の変数のためにＶＣＢへ入力フィールドをマップして戻さなければならないＰＯＳＴ－ＷＡＩＴルーチンとして実行させることを含む。これはデータを一時記憶を介して通過させることを含む。

【０１５９】その後、入力が分析され、入力データがすべての関連の変数に挿入される。すべての入力フィールドについて５４８でフィールドの正当性が立証される。それゆえ、多数のフィールドについて数字だけが入れられることを確実にするようにチェックが行なわれてもよい。同様に、入力フィールドが特定の属性を有するか否かを決定するチェックが行なわれてもよい。

【０１６０】その後、ブロック５５０でフィールドのいずれかでエラーがあったか否かを決定するチェックが行なわれる。もしエラーがあれば、エラーメッセージが発生され、カーソルが誤ったフィールドに位置決めされ（５５２）、その後ルーチンが５４０へ分岐して戻り、エラーメッセージ表示を発生する。

【０１６１】もし５５０のチェックで特定フィールドのエラーを明らかにすることができなければ、さらなるチェックが行なわれ、フィールドが脈絡において正しい（たとえば２つの隣接するフィールドが個々には正しいが、エラー条件がフィールドの組合せに関して規定されるかもしれない。）という相互確証が５５４で行なわれる。相互確証に基づき、フィールドが脈絡のエラーを含むか否かに関する決定が行なわれる。もしなければ、５５８で呼出人への復帰が行なわれる。もし脈絡のエラーがあれば、エラーメッセージがブロック５５２に従って発生される。

【０１６２】個々のフィールドの双方の確証は完全にプログラムの制御下にあることが注目されるべきである。様々な仕様、利用ルーチンおよび共通の状況の取扱いの簡略化への便宜があってよいが、一般にいかなる可能性のある正当性立証も可能である。フィールドの相互の正当性立証がより意味的な関心を含んでもよく、したがって特別なプログラミングを必要とする傾向がある。

【０１６３】図２９は時間遅延ルーチンによって行なわれる動作のシーケンスを示す。時間遅延機能は予め定められた時間間隔で起きる（wake up)ために使用されてもよく、入ってくる電子郵便が到着したか否かを確かめるチェックを行ない、それ自身をそのメイルに取付け、それによって入ってくる電子データ変換を効率的に取り扱うために使用されてもよい。したがって、このような時間遅延機構を通して、移動プログラムはメイルが到着し

たかどうかをチェックするために特定のメイルボックスを予め定められた時間間隔で検査し得る。もしメイルが到着していれば、移動プログラムはさらなる受け手によって取扱われるべき行先へメイルを送ることができる。代替的に、移動プログラムは（メイルのような）入ってくるデータを検査し、様々な内容のインジケータに基づき自動的にトラバースを行ない、メイルを適切に処理し得るそれ自身の新しい「瞬間」を引き起こすことができる。むろん、最初の「瞬間」は到着するあらゆる瞬間に実行および処理を継続し得る。

【０１６４】たとえば、もし入ってくる情報がたまたまＥＤＩトランザクションであったならば、移動プログラムは（たとえばＲＥＡＤビルトイン機能を使用して）情報を読出し、それを壊して離し、内部変数にし、誰によって処理されるべきかを決定し、適切なトラバーサルを行なう。一度うまく送られれば、レターが処分、移動、または記録され、プログラムはその変数をクリアし、入力をさらに探すことを再開することができる。

【０１６５】代替的に、到着した材料の型を決定した後、別のプログラムを呼出し、入ってくるデータを処理することができる。もし他方のプログラムがたまたま移動プログラムであれば、そのプログラムは必要な入力情報を与えられ、それ自体を取扱いに適切なようにＴＲＡＶＥＲＳＥし得る。

【０１６６】これはたとえば１つの移動プログラムがＥＤＩトランザクションのような入来データのための自動ルータとして作用し、それ自体を取り扱う準備ができていないトランザクションを他の移動プログラムに渡すことを可能にするであろう。

【０１６７】さらに、もしＥＤＩがサインされれば、その移動プログラムは即座に署名を確証することができるであろう。もし署名が有効であれば、特にそれが米国特許第5,005,200 号に従って行なわれれば、内容に対する許可がプログラムに基づいて画面表示され、移動プログラムは自動的にインスタンスをスピンオフして、入来トランザクションを処理することができるであろう。

【０１６８】たとえば、適切な高められた許可があれば、入ってくる買い注文は自動的にかつ瞬時に発送部に送られ、注文に応じることが可能であろう。

【０１６９】到着したがサインされていない品物、または権限のある署名よりもむしろ単純な署名を使用した品物であれば、特例処理またはより詳細な検査のために様々な事務員に送られるであろう。

【０１７０】ブロック５７０に示されるように、時間遅延ルーチンは指定された時間に対してシステムアラームクロックをセットする。その後、補助記憶へのデータのオプションのロールアウトが適切なルーチンでＰコード割込をスケジューリングすることによって実行され（５７２）、その後指定された時間期間の経過後にデータのロールインが実行される。その後呼出ルーチンへの戻りが発生する（５７６）。

【０１７１】図３０は「ディレクトリから選択」機能のための一連の動作を示す。このディレクトリはファイルのディレクトリまたはユーザのディレクトリなどであり得る。はじめに、リストがすべての候補項目から５８０で作成される。その後ディスプレイは５８２でリストの少なくとも一部分を表示するように発生される。ユーザは示されたこれらの項目から選択する機会を有し（５８３、５８５）、その後この機能は機能結果としてかまたは特別な変数の組のいずれかとして、選択された項目の名前を戻す（５８４）。

【０１７２】再び他で説明されたように、実際のＷＡＩＴはＰコード割込機能の使用によって実行される。この場合、ＩＮＴＥＲ－ＲＯＬＬＯＵＴルーチンはユーザが選択肢から選択するのを待ち、入力をＰＯＳＴ－ＷＡＩＴルーチンを介してプログラム変数に戻す。

【０１７３】図３１はインタプリタプログラムがユーザがディジタル署名を実行することをどのように許容するかを示すルーチンである。ブロック６００に示されるように、ディジタル的に署名されるべきデータはプログラムがアクセスすることが可能なデータに基づいて組立てられ、これにはユーザが供給した入力、ファイルから読出されたデータ、前のトラバーサルから蓄積されたデータ、ユーザの環境に基づくデータ（たとえばユーザのＴＳＯ名）、時間、プログラムそれ自体に組込まれたデータ、およびビルトイン機能から引出されたデータ（たとえばビルトインＸ１２データディクショナリ）が含まれる。適切な情報がユーザに表示される（６０２）。ユーザはそれから、ブロック６０４に示されるように、自分がデータにサインしたいかどうかを決定する。もしユーザが署名を実行したいことを示せば、システムは、図２６に例示されるように、署名機能を呼出し、さらにユーザと対話し、署名を完了する（６０６）。その後６０８でディジタル署名が発生されプログラム変数として保管される。

【０１７４】図３１およびそれに続くフローチャートは、どのようにユーザがそこに説明された移動プログラム法を利用しながら、比較的少ない動作を行なって、前述のインタプリタに組込まれた強力な機能を達成するかをある程度示す。

【０１７５】図３２はユーザがどのように受信された情報を確証するかを例証する。ブロック６１０に示されるように、確証されることが期待されるデータが組立てられる。その後、組立てられたデータおよび保管されたディジタル署名とともに任意の可能な権限要求を有する「確証」機能が呼出される。確証機能は米国特許第5,005,200 号で述べられたように、または従来のディジタル署名動作が変数をサインするために利用される場合には標準的なディジタル署名技術を使用して、達成され得る。その後、ブロック回路１２の処理に基づいて署名が

確証されたかどうかが決定される（６１４）。もしそうであれば、プログラム実行が続行される。もしそうでなければ、誤り状態は６１６でそのデータが不当にも変更されたか、または何からの種類のプログラミングエラーがあったことを示すことになる。戻りコードは署名が無効であるかどうか、それが許可能力を支援したかどうか、およびもしそうであればその許可が確認されたかどうかをプログラムが区別することを可能にするように規定される。

【０１７６】図３３は移動プログラムがどのように転送されるべきファイルを集めるかを例示す。はじめに、プログラムは６２０でたとえばユーザにファイルのリストを表示することによって転送されるべきファイルを決定する。ファイルを決定するためにユーザ対話をする必要があるかどうか決定するためにチェックが行なわれ得る（６２２）。もしイエスであれば、ユーザは６２４で転送されるべきファイルを決定するように促される。もしファイルを決定するためにユーザ対話を行なう必要がなければ、全体のファイル内容は６２６で転送されるべきデータの組に付けられる。動作は図２２に述べられた付加された機能を使用して達成され、この図は前に説明されたようにファイルコントロールブロックを構築することを含む。

【０１７７】図３４は指定されたファイルからデータを読出す際に実行される移動プログラム動作を例示する。はじめに読出されるべきデータを含むファイルが決定される（６３０）。その後、６３２でデータが指定されたファイルから読出されプログラム変数として保管される。図３５は移動プログラムがどのようにプログラム変数からファイルを更新または作成し得るかを例示する。ブロック６４０に示されるように、その中にデータが書込まれるべきユーザファイルがまず決定される。その後、６４２でプログラム変数をユーザファイルに書込む機能が呼出される。

【０１７８】すべての場合に明らかに説明されていなくても、データ損失、変更、損傷または露見につながり得るプログラム機能であれば何でも、セキュリティコントロールの支配下にあることが理解されなければならない。このようなコントロールはプログラムレベルで与えられるかまたは入来プログラムに結びつけられ、恐らくはやはりユーザによって課せられるものと或る予め定められた態様で組み合わされることによって行なわれる。

【０１７９】したがって、たとえば上述の場合には、移動プログラムはもしプログラムがそのように権限づけられればユーザのデータファイルを読出すかまたは書込むことができるだけである。

【０１８０】セキュリティ制約は少なくとも以下のクラスの機能に対して存在する。データをユーザに表示する。

【０１８１】入力をユーザから請求する。ディジタル署名を実行する。

【０１８２】ユーザファイルからデータを読出す。ユーザファイルを作成する。

【０１８３】ユーザファイルを消去する。データをユーザファイルに書込む。

【０１８４】ユーザファイル名を変える。ユーザファイルを付加する。

【０１８５】付加されたファイルをユーザファイルに出す。ディジタル公証装置を呼出す。

【０１８６】入来する電子メイルを受信する。電子メイルの内容を読出す。

【０１８７】入来するメイルを移動または記録する。入来するメイルを消去する。

【０１８８】外部へ行く電子メイルを作成する、または様々なタイプのデータ送信を行なう。

【０１８９】様々なタイプの装置、デバイスおよびサービス（ＦＡＸ、プリンタ、オフィス機器、ロボット装置、製造装置など）に結合される。

【０１９０】プログラムトラバーサルを実行する。外部プログラムを呼出す。

【０１９１】他の移動プログラムをアクセス、更新、能動化、消去、変更、呼出し、または付加する。

【０１９２】図３６はどのように移動プログラムが分割され多数の異なった受信者に送られるように設計され得るかを例示し、図３７はどのように前に分割されたプログラムが併合され得るかを示す。

【０１９３】まず図３６を参照して、移動プログラムはたとえば多数の異なった受信者からサーベイデータを獲得するために、またはデータを集めるもしくは組織の多数の異なった幹部にデータを分配するために、移動プログラムを分割する必要があるかもしれない。はじめに、移動プログラムは６５０で分割の準備をするために様々なハウスキーピング動作を実行する。その後、変数は６５２で特定のアプリケーション要求、たとえば特定のユーザによって実行されるサーベイに従って設定される。６５４で行先ユーザが決定され、図２０および図２１に従ってトラバース機能が呼出され、プログラムのイメージ、プログラム変数とともに個々の受信者に適合された任意の他の適切なデータを送信する。送信された変数はインスタンス１（６５６）からインスタンス２（６５８）、インスタンス３（６６０）、そしてインスタンスＮ（６６２）へと変化し得る。

【０１９４】それに対して送信すべきより多くの行先があるかどうかを決定するためにチェックが最終的に行なわれる（６６４）。もしそうであれば、ルーチンは６５２に分岐して戻り、さらなる行先に送信する。もしさらなる行先がなければ、最終的な転送が６５４に対して上で説明されたのと同一の態様で６６６で実行され、６６８で最終的な「インスタンス」をもたらし、その後分割動作の完了になる。

【０１９５】他の例において、マスタプログラムが単に何らかの他の処理に進むこともあり得る。恐らく、もしそれが入力ディストリビュータのようなバッチ環境で実行していれば、かつすべての入力が現在使い尽くされていれば（多数のユーザにスピンオフされたばかりで）、それは何か他のものが到着するまで遅延するであろう。

【０１９６】図３７の併合動作を参照して、移動プログラムはそれ自体をユーザからユーザに転送して、併合動作が完了するまでデータをさらに併合するインテリジェンスを有する。はじめに、移動プログラムは併合行先に到着し実行される（６８０）。これが予め定められた変数が設定されることによって決定されるマスタ「インスタンス」であるかどうかを決定するためにチェックが行なわれる。もし６８２でこれがマスタインスタンスでないことが決定されれば、６８４でスレーブインスタンスが識別される。（６８５）で、スレーブプログラムはそれが特別な「ＤＥＢＲＩＥＦ」パラメータで呼出されたどうかをチェックし（これはいつスレーブがマスタによって呼出されているかを決定するためにこのプログラムによって使用される単なる規約である）、もしそうなら（６８７）、すべての適切な情報はマスタインスタンスに戻り、退出する。もしこれがＤＥＢＲＩＥＦ呼出しでなければ、マスタインスタンスが利用可能であるかどうか、つまり既に到着しているかを決定するために６８６でチェックが行なわれる。もしマスタインスタンスが利用可能であれば、図１８に示される呼出の使用によって、６９６でマスタインスタンスへの呼出が行なわれる。マスタインスタンスが呼出された後、ルーチンはブロック６８０に分岐して戻る。もしマスタが利用可能でなければ、６８８でその連続に対するマスタコントロールがまだ到着していないというメッセージが発せられる。

【０１９７】マスタインスタンスが到着し、呼出されたと仮定して、ブロック６８２で、これがマスタインスタンスであることが決定され、６９２で何か他のスレーブインスタンスが到着したかどうかを決定するためにチェックが行なわれる。もしそうであれば、スレーブインスタンスは予め定められたパラメータで呼出され、６９４でデータの収集を開始する（恐らく「デブリーフィング(debr ief ing)」と呼ばれる）。エントリポイントＡでデータはインスタンスから集められ、７０６でマスタに戻され、収集ファイルに書込まれる。その後呼出されたばかりのインスタンスは７０８で消去され、ルーチンは６９２に分岐して戻り、その場合には他のインスタンスが到着したかどうかのさらなる情報が集められる。

【０１９８】もし他のさらなるインスタンスが到着していなければ、そのファイルはすべてのインスタンスがすべて到着したかどうかを見るためにチェックされる（６９８）。もしそれらが到着していれば、７００で決定されたように、データは収集から移動プログラムの変数に読込まれ得る。収集ファイルの予想されるサイズおよび処理の性質に依存して、その瞬間に完成されたファイルを処理し、それ自体を次の行先にトラバースするか、またはその結果を大切に保護して単純なメッセージ、恐らくはＥＤＩトランザクションにし、その生のデータを単に送信することが、マスタプログラムにとってより望ましいかもしれない。

【０１９９】他の場合において、プログラムがファイルをそれ自体にＡＴＴＡＣＨし、それを大量に別のプロセスに転送することが適切であるかもしれない。７０４でそのファイルは消去され、集合体データは次の行先に送信される。もしすべてのインスタンスがまだ到着していなければ、「形式が到着するのを待つ」のようなメッセージが発行され（７０２）、ルーチンは一時的に退出させられる。

【０２００】図３８は以前に分割された移動プログラム情報を併合するための代替のアプローチを示す。ブロック７１０で示されるように、移動プログラムは併合行先に到着し、実行される。収集されたデータは７１２で特別ファイルに書込まれる。７１４に示されるように、すべての他のインスタンスが到着したかどうかを決定するためにチェックが行なわれる。もしそうであれば、収集されたデータは７１６で処理され、プログラムは７１８で次の行先にトラバースし、ルーチンは退出させられる。もしすべての他のインスタンスが７１４で決定されたように到着していなければ、「より多くの形式が到着するまで待つ」のようなメッセージが表示され（７２０）、現在のインスタンスは７２２で消去され、ルーチンは退出させられる。

【０２０１】図３９は移動プログラムが電子データ交換（ＥＤＩ）生成機能を収容するためにどのように設計されたかを示すフローチャートである。図３９は特定の「Ｘ１２」標準特徴がどのように使用され得るかをより具体的に示す。Ｘ１２標準は関連するデータディクショナリおよびセグメントディクショナリを有する。Ｘ１２セグメントディクショナリは、たとえば、買い注文を規定するために必要なすべてのセグメントを規定するために使用され得る。各セグメントはその後ディクショナリで調べられる一片のデータであるとして規定される。項目の量を特定するために多くの異なった方法があるので、データの多くの変化がＸ１２で許容される。

【０２０２】このシステムはＸ１２データディクショナリをビルトイン機能と呼ばれ得るインタプリタにはめ込む。ブロック７２０で示されるように、はじめセグメント名前および項目「ＸＸ、ＹＹ、ＷＷ、．．」を特定することによってＸ１２サブルーチンへの呼出が行なわれる。プログラムは組織の環境に典型的な人気のある共通オプションにＸ１２データコードを与えることが可能であり、その結果通常の使用のためにオプションの短いリストを構築することが可能である。このような項目の例

は買い注文脈絡において、項目数、部品数および量である。この呼出はビルトインデータディクショナリへの呼出になる。

【０２０３】その短いリストが空であるかどうかを決定するためにチェックが行なわれる（７２４で示されるように）。もしそうであれば、セグメント名が７３６ですべての関連するデータオプションのセグメントディクショナリテーブルを場所決めするビルトイン機能Ｘ１２ＳＥＧＬＩＳＴを呼出すために使用される。その後、Ｘ１２ＤＡＴＡＮＡＭＥビルトイン機能はデータディクショナリを使って７３８で各関連する説明データを拡張するために使用され、長い完全なリストが７４０で表示される。

【０２０４】もし７２４でのチェックが短いリストがあることを示せば、Ｘ１２ＤＡＴＡＮＡＭＥデータディクショナリは短いリストのオプションの各々の拡張された記述を場所決めするために使用される。その後７２８で短いリストが表示される。その後ユーザが７３０で示されたような完全な長いリストを望んでいるかどうかを決定するためにチェックが行なわれる。もしその答えがイエスであれば、ブロック７３６は上述のように実行される。もしノーであれば、短いリストかまたは長いリストのいずれかからのユーザの選択が受け入れられる（７３２）。

【０２０５】すべてのデータが集められるかどうかを決定するためにブロック７３４でチェックが行なわれる。もしそうであれば、我々は７４２で完成されたＸ１２トランザクションを組立てかつ発行し、それからそのルーチンを退出させる。７４２に関連して参照される発行動作に対して、この発明は全体の移動プログラムをメイルすることに加えて、特定の組のＸ１２データをメイルする能力を熟慮する。もしすべてのデータが７３４のチェックによって示されるように集められなければ、より多くのデータ項目が検索され、ルーチン実行が繰返される。

【０２０６】図４０は電子データ交換トランザクションを受信する際の移動プログラムの用途に関連する。たとえば、特定のユーザは移動プログラムが発生した買い注文を受信したかもしれない。はじめに、受信されたＥＤＩトランザクションは７５０で読出される。恐らくタイマ遅延によって、図２９を使って説明されたように、入力としてそれ自体のコピーを産出する移動プログラムが到着する。符号化されたＥＤＩはそれから７５２でプログラム変数に構文解析される。受信されたＥＤＩはそれから記録保管所に移動され、可能な監査のために受信されたものを保存する。このセグメントは７５６で結合されたセグメントディクショナリを経て処理される。Ｘ１２に関連するセグメント規則は強化され、それはたとえば７５８で特定のフィールドの或る種類のデータを有しないことに関連するかもしれない。各データ項目に対して、各セグメントに関連するデータディクショナリは７６０で場所決めされる。ＤＥＳＣ＝Ｘ１２ＤＡＴＡＮＡＭＥ（ＳＥＧＣＯＤＥ，ＤＡＴＡ　ＩＴＥＭ）である７６２で示されるようなステートメントに対して、このステートメントはデータディクショナリへの呼出という結果になり、データ項目の意味のある記述を得るであろう。検索された意味のある記述は表示変数におかれ、その結果たとえば買い注文フォーマットでの買い注文の表示になる。すべてのデータ項目はブロック７６２に分岐して戻ることによって処理され、すべてのセグメントは７５６に分岐して戻ることによって処理される。

【０２０７】この好ましい実施例はまたディジタル公証、本発明者の米国特許第5,001,752号（引用によりここに援用する）によって説明されるような公証装置、または同様に他の装置にアクセスすることが可能なビルトイン機能を与えることによってディジタル公証機構へのアクセスを許容する。

【０２０８】移動プログラムがそのような機構にアクセスすることを許容することによって、移動プログラムはデータをプラットホームに移動させることが可能であり、そこではディジタル公証は容易にアクセスされ、そのようにするためにビルトイン機能を使用する。これは重要な署名、入ってくるトラフィックのための、または何らかの他の理由のためのタイムスタンプのための公証を可能にする。このような公証はプログラムの厳しい制御下にあるので、自動的であろうとユーザ要求に基づくものであろうと任意の基準が使用され得る。

【０２０９】やはり以前に説明されたように、この機構は外へ出ていくＦＡＸへの結合を考慮しており、その結果電子形式は、ＥＤＩに変換されるまたは印刷されることに加えて、最終的な受信者にＦＡＸすることも可能であるようにされる。

【０２１０】また明らかに述べられていないが暗に示されているように、移動プログラムがＥＤＩトランザクションを発しているときでさえ、それは依然として後で能動化されてもよい。一例はまずはじめに電子要求として機能し、それから十分な是認署名の後、買い注文を発生する移動プログラムであろう。それはその後それ自体を保管所に送り、そこで対応するインボイスおよび請求書が最終的に到着（電子的または他の態様で）したときに再び能動化され、その注文を受け取った船積および請求書と一致させるための方法として機能し得る。それはどの項目が受け取られ、どれがまだペンディングであるかを追跡する論理を組込み得る。それ自体を柔軟に方向づける能力のために、それは多くの異なった場所に広がり得る。船積および受取りを扱う限りにおいて、移動プログラムをバーコードリーダと結合し、人間のデータ入力なしに送りかつ受け取られる材料を実証することもまた可能である。

【０２１１】好ましい実施例は移動プログラムがオフィ

ス機器および他の装置ならびに機構を含む様々な装置に結合され得ることを想定する。

【０２１２】また、任意の所与のトラバーサルは様々な受信者に同時に送られ得る。以下のリストは好ましい実施例が実行することが可能な上述の機能の多くを繰返しかつ要約する（かつ幾つかの付加的な機能を識別する）。このリストは例示のみであり、この発明が有利に適用され得る他の多くのアプリケーションを余すところなく示すことが意図されるものではない。レイアウト言語（たとえばＴｘＸまたはＳＣＲＩＰＴに類似）を使用してユーザにデータを表示する。

【０２１３】レイアウト型言語（ＴｅＸまたはＳＣＲＩＰＴに類似）を使用してユーザから入力を請求する。

【０２１４】プログラムコントロール下で計算されたデータのためのディジタル署名を実行する。

【０２１５】プログラムコントロール下で計算されたデータに基づいてディジタル署名を確証する。

【０２１６】署名者の証明から引出された提案を送ることを恐らく含んで共同署名を処理する。

【０２１７】ユーザファイルからデータを読出す。ユーザファイルを作成する。

【０２１８】ユーザファイルを消去する。データをユーザファイルに書込む。

【０２１９】ユーザファイル名を変える。入来する電子メイルを受信する。

【０２２０】電子メイルの内容を読出す。入来するメイルを移動または記録する。

【０２２１】入来するメイルを消去する。外へ行く電子メイルを作成する。

【０２２２】外へ行くＦＡＸサーバーに結合するおよびそれを制御する。プリンタに結合するおよびそれを制御する。

【０２２３】グラフィカルイメージを作成する。オーディオ信号を送受信することが可能なデバイスに結合するおよびそれを制御する。

【０２２４】オフィス機器、コンピュータ機器（テープ、ディスクなど）、ロボット装置、製造装置などを含む様々なタイプの装置にアクセスする。

【０２２５】移動プログラムのインスタンスを多重トラバーサルによって幾つかのインスタンスに分割する。

【０２２６】恐らく同一のプログラムを示すことさえせずに、幾つかの移動プログラムに含まれたデータを単一の形式に再結合することが可能である。

【０２２７】移動プログラムの他のインスタンスを消去する。外部プログラムを呼出す。

【０２２８】サブルーチンとして他の移動プログラムを呼出す。独立して実行する機能として他の移動プログラムを能動化する。

【０２２９】ドーマント（非実行）移動プログラムからデータを抽出する。プログラムの名前、他の状態などのような、それを実行する必要性を伴わずに他の（非実行）移動プログラムについての情報を決定する。

【０２３０】ディジタル署名に関連する証明から情報を抽出する。この情報はもし共同署名要求が含まれれば直接送ることを助けるために使用される。

【０２３１】他のプログラム内のデータ変数として移動プログラムのコピーをつくる、またはファイルとして他のものへ移動プログラムをＡＴＴＡＣＨする。

【０２３２】１つの移動プログラム（「キャリア」）を他のものの新しいバージョンを様々な行先に輸送するために使用し、現存するインスタンスのプログラムセグメントをそのプログラムのより最新のバージョンである別のものと置換える。こうするための１つの方法は、より新しいプログラムセグメントが現存する移動プログラムの端部に加えられるようにすることである。現存するインタプリタ／ローダの増強は、クロージャセグメントに従うプログラムセグメントは提案されたプログラム改訂を反映したことを認識するであろう。どんな通常送信でも実行された後に、提案された改訂されたプログラムに関連するディジタル署名を実証し、もしそれらが適切な権限を実行していれば、標準トラバーサルの一部として到着したプログラムの代わりに新しいプログラムを使用し始めるであろう。

【０２３３】ユーザファイルを付加する。付加されたファイルをユーザファイルに出す。

【０２３４】前に付加されたファイルを取り除く。ディジタル公証装置にアクセスする。

【０２３５】プログラムトラバーサルを実行する。ユーザデータを送信し（トラバーサル以外で）、その結果その送信は移動プログラムそれ自体を含まないようにする（たとえばメッセージを他の行先に単に送る）。

【０２３６】ＥＤＩ（たとえばＸ１２またはＥＤＩＦＡＣＴ）の使用、作成、表示、構築および受信を単純化するためのビルトイン機能を使用して、移動プログラムのこれらの機能を供給する必要性を伴わずに、共通の情報および機構を便利に供給する。これはデータエレメントディクショナリ、セグメントディクショナリ、セグメント規則およびトランザクションセット自体にアクセスするビルトイン機能を含む。

【０２３７】この発明を現在最も実際的な実施例であると考えられるものと関連して説明してきたが、この発明は開示された実施例に制限されるものではないことが理解されなければならず、それとは反対に前掲の特許請求の範囲の精神および範囲内に含まれる様々な修正および等価の配列をカバーすることが意図される。

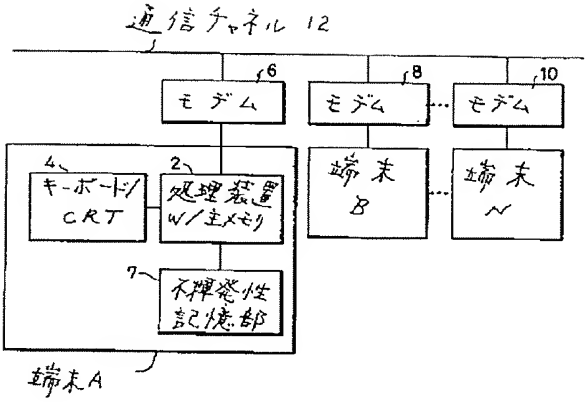
【図面の簡単な説明】

【図１】本願発明の例示的実施例に従う通信システムのブロック図である。

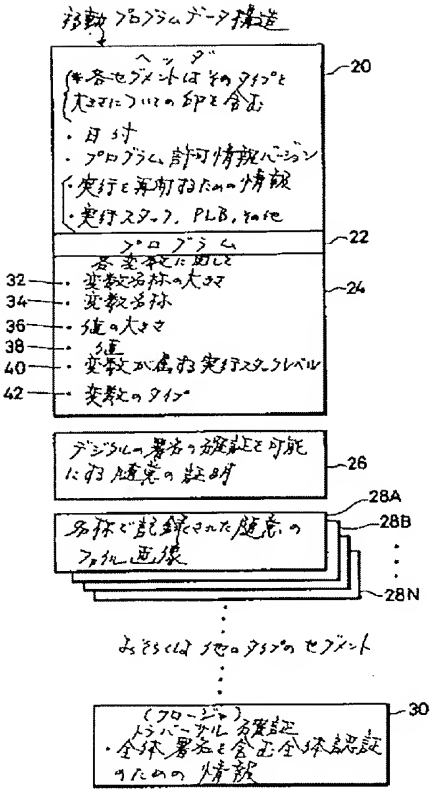
【図２】移動プログラムとその関連するコンポーネントの例示的構造を示す図である。

【図３】例示的実行制御領域データ構造を示す図である。
【図４】移動プログラムがファイルをそれ自体に取り付けまたはそれ自体からファイルを外す際に使用されるファイル制御ブロック（ＦＣＢ）のデータ構造を示す図である。
【図５】移動プログラムの実行の際に使用される処理制御ブロックを示す図である。
【図６】変数を制御するために使用される変数制御ブロックデータ構造（ＶＣＢ）を示す図である。
【図７】例示的移動プログラムローダの図である。
【図８】ヘッダがロードされる様子を示す図である。
【図９】移動プログラムの「プログラム」セグメントがロードされる様子を示す図である。
【図１０】移動プログラムの「変数」セグメントがロードされる様子を示す図である。
【図１１】移動プログラムの「証明」セグメントがロードされる様子を示す図である。
【図１２】移動プログラムの「ファイル」セグメントがロードされる様子を示す図である。
【図１３】移動プログラムの「クロージャ」セグメントがロードされる様子を示す図である。
【図１４】Ｐコード命令の処理の際に行なわれる動作を表わす図である。
【図１５】Ｐコード動作が行なわれた後に行なわれる処理を示す図である。
【図１６】（Ａ）および（Ｂ）はプログラム規定による機能または呼出を取り扱うための処理を示す図である。
【図１７】内蔵機能を取り扱うための動作のシーケンスを示す図である。
【図１８】外部機能または呼出を実行するために行なわれる動作のシーケンスを示す図である。
【図１９】外部機能または呼出を実行するために行なわれる動作のシーケンスを示す図である。
【図２０】移動プログラムがそれ自体を予め定められた受け手に対しメイルする際に行なわれる動作を示す図である。
【図２１】移動プログラムがそれ自体を予め定められた受け手に対しメイルする際に行なわれる動作を示す図である。
【図２２】移動プログラムに対しファイルを取り付けるための動作のシーケンスを示す図である。
【図２３】ファイルがユーザのシステムから消去され得る様子を示す図である。
【図２４】移動プログラムからファイルを取り外す際に行なわれる動作のシーケンスを示す図である。
【図２５】ファイルがユーザのファイル内に変換されたときに行なわれる動作のシーケンスを示す図である。
【図２６】資料がディジタル的にサインされる場合に行なわれる動作のシーケンスを示す図である。
【図２７】「ＩＮＴＥＲ－ＲＯＬＬＯＵＴ」機能により行なわれる動作のシーケンスを示す図である。
【図２８】情報をユーザに対し表示する際に行なわれる動作のシーケンスを示す図である。
【図２９】「時間遅延」ルーチンにより行なわれる動作のシーケンスを示す図である。
【図３０】「名簿からの選択」機能のための動作のシーケンスを示す図である。
【図３１】インタプリタプログラムによってユーザがディジタル署名を行なうことが可能になる様子を示すルーチンを示す図である。
【図３２】受け取られた情報をユーザが確証する様子を示す図である。
【図３３】移動プログラムが伝達されるべきファイルを収集する様子を示す図である。
【図３４】特定されたファイルからデータを読出す際に行なわれる移動プログラムの動作を示す図である。
【図３５】移動プログラムがプログラム変数からファイルを更新またはつくり出し得る様子を示す図である。
【図３６】移動プログラムが分割されるべく設計されかつプログラムを数々の異なる受け手に対し送り得る様子を示す図である。
【図３７】事前に分割されたプログラムがマージされ得る様子を示す図である。
【図３８】事前に分割された移動プログラム情報をマージする代替的方策を示す図である。
【図３９】移動プログラムが電子文書交換発生機能を受け入れるべく設計された態様を示すフローチャートである。
【図４０】電子データ交換トランザクションを受ける際の移動プログラムの使用について示す図である。
【符号の説明】
２…処理装置Ｗ／主メモリ
４…キーボード／ＣＲＴ
６…モデム
１２…通信チャネル

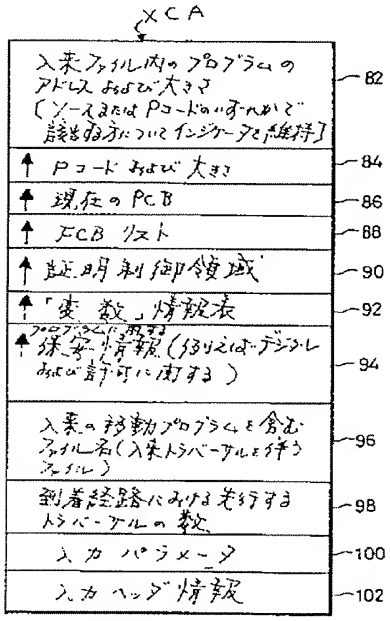
【図１】



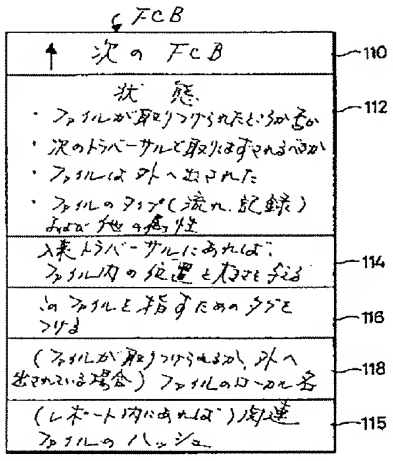
【図２】



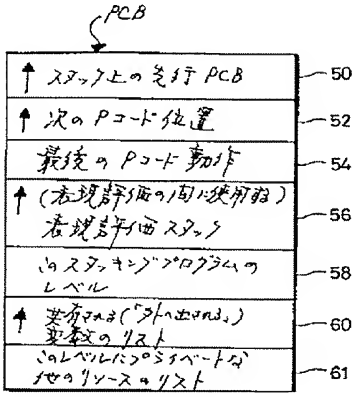
【図３】



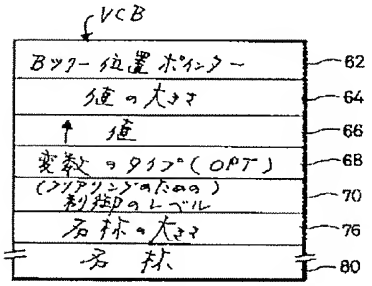
【図４】



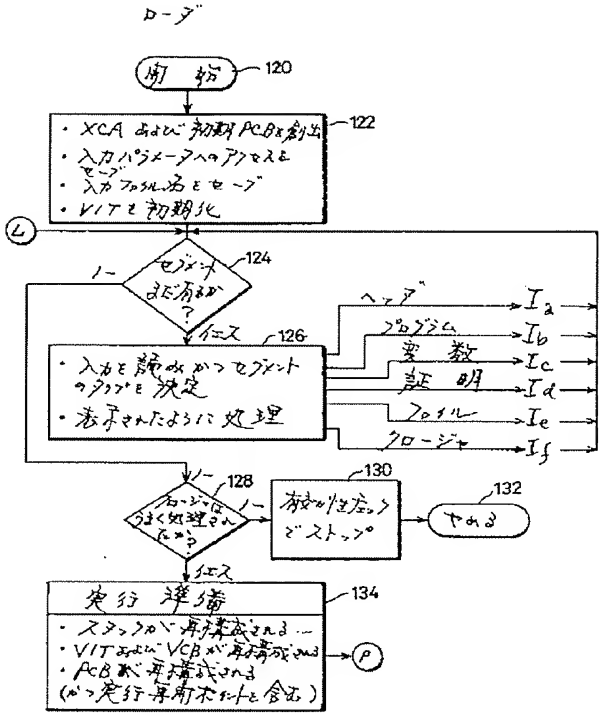
【図５】



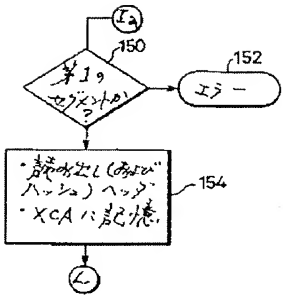
【図６】



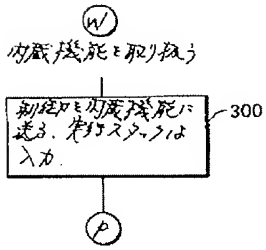
【図７】



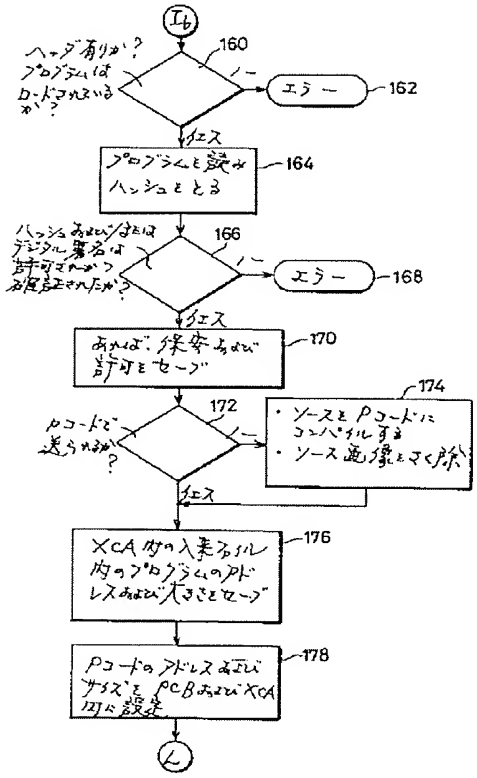
【図８】



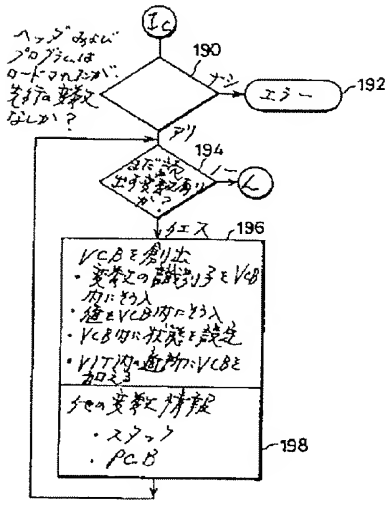
【図１７】



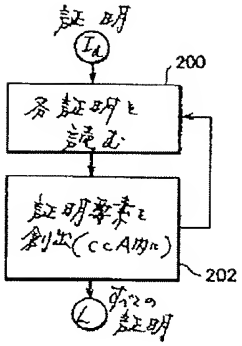
【図９】



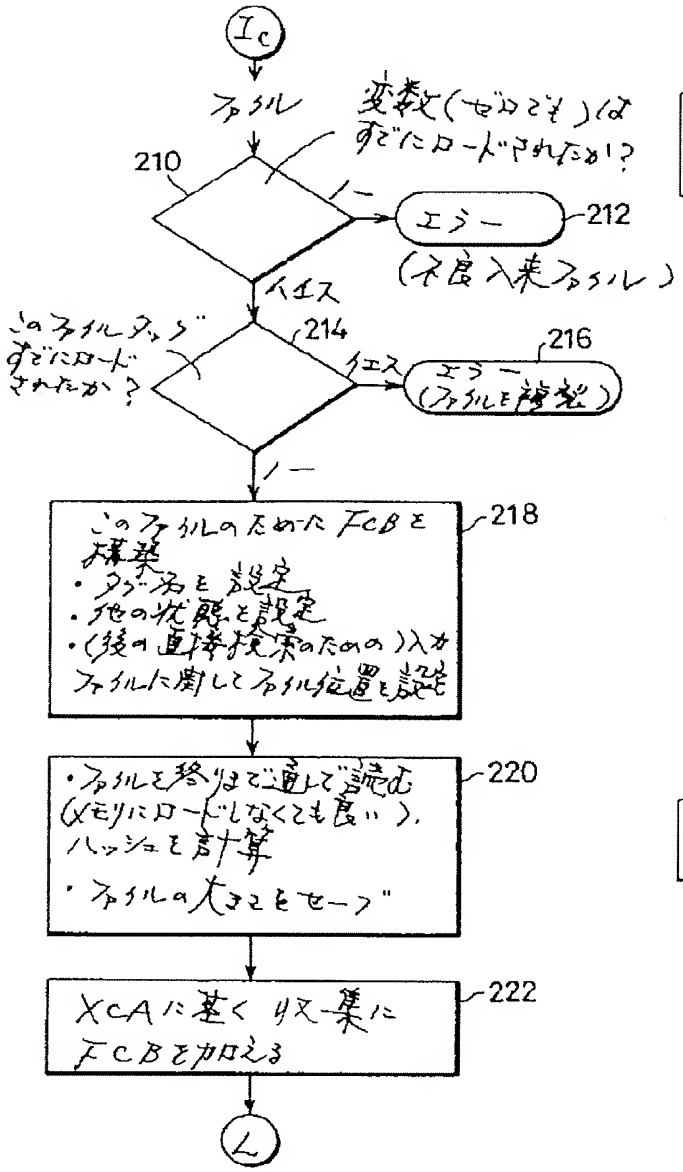
【図１０】



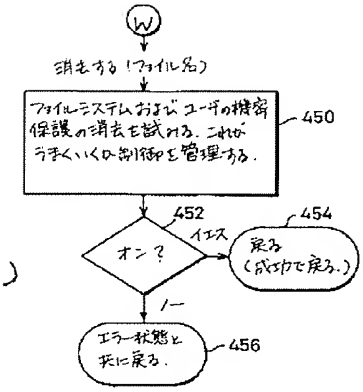
【図１１】



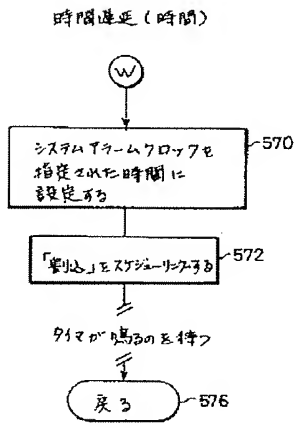
【図１２】



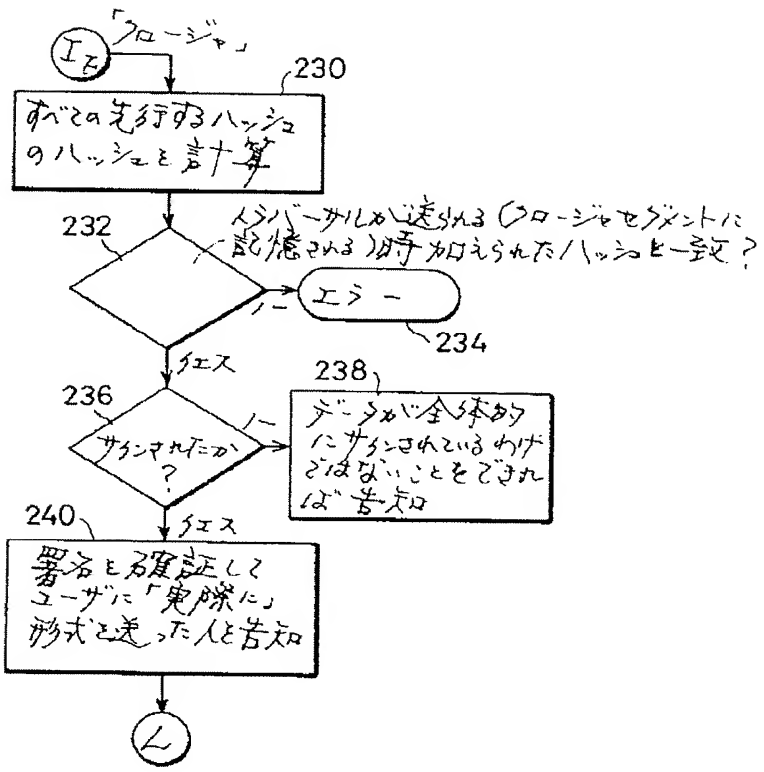
【図２３】



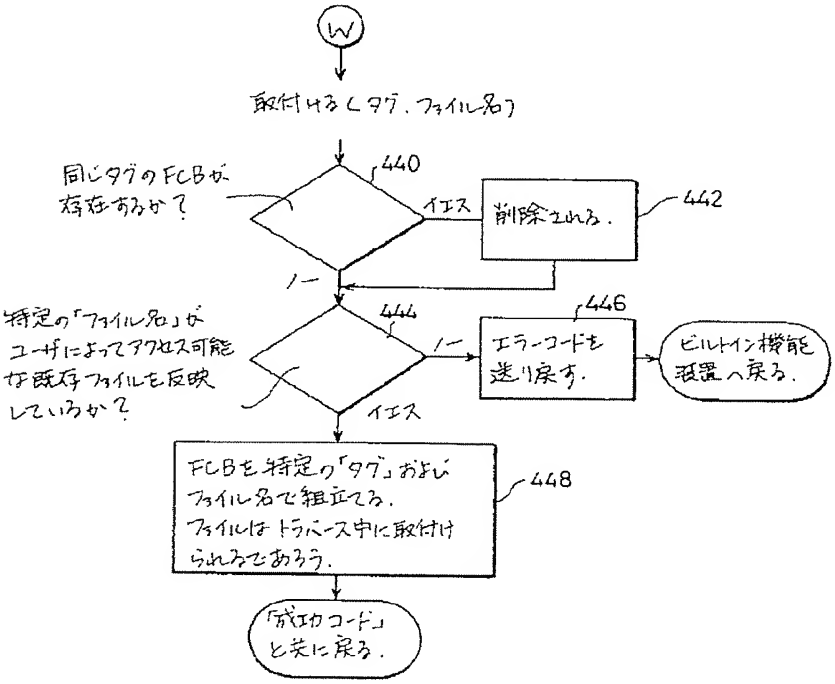
【図２９】



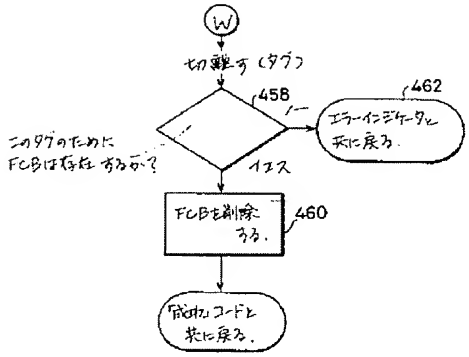
【図１３】



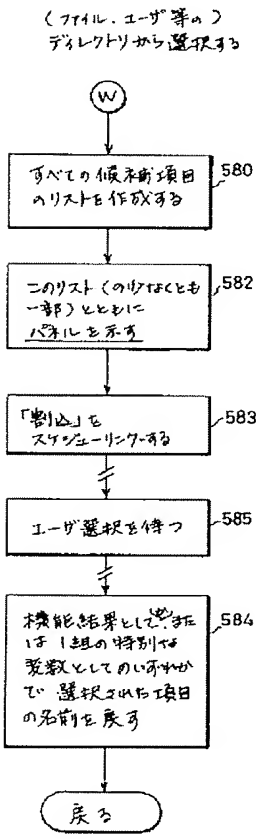
【図２２】



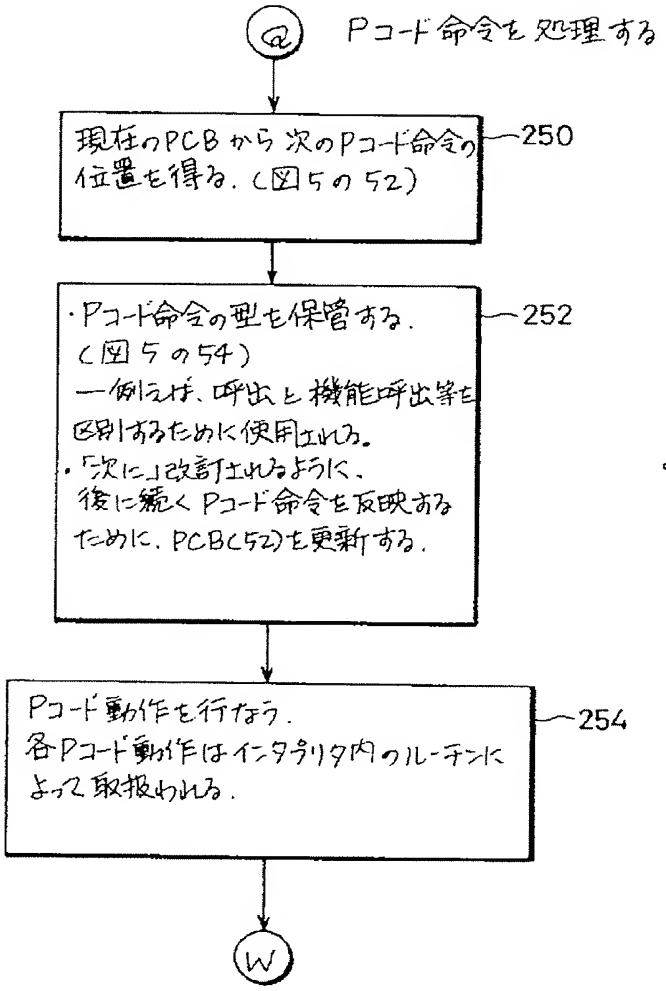
【図２４】



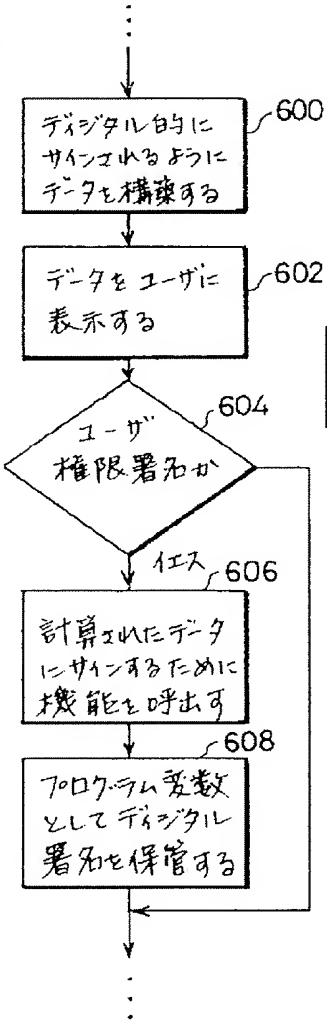
【図３０】



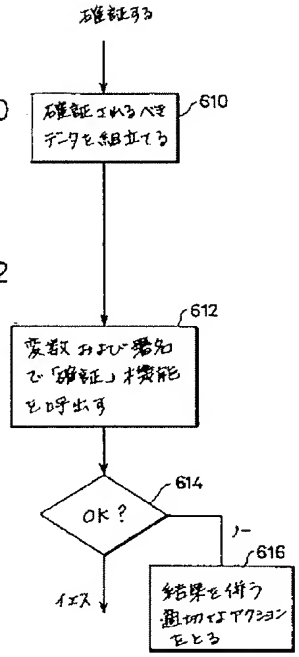
【図１４】



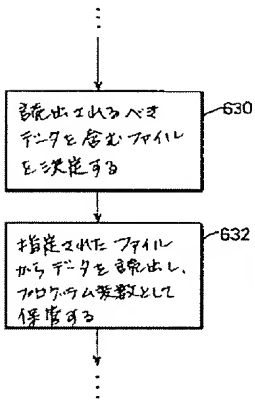
【図３１】



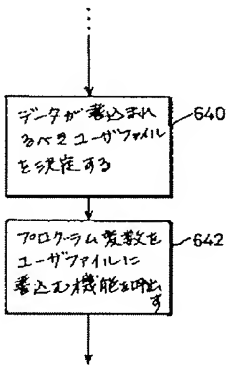
【図３２】



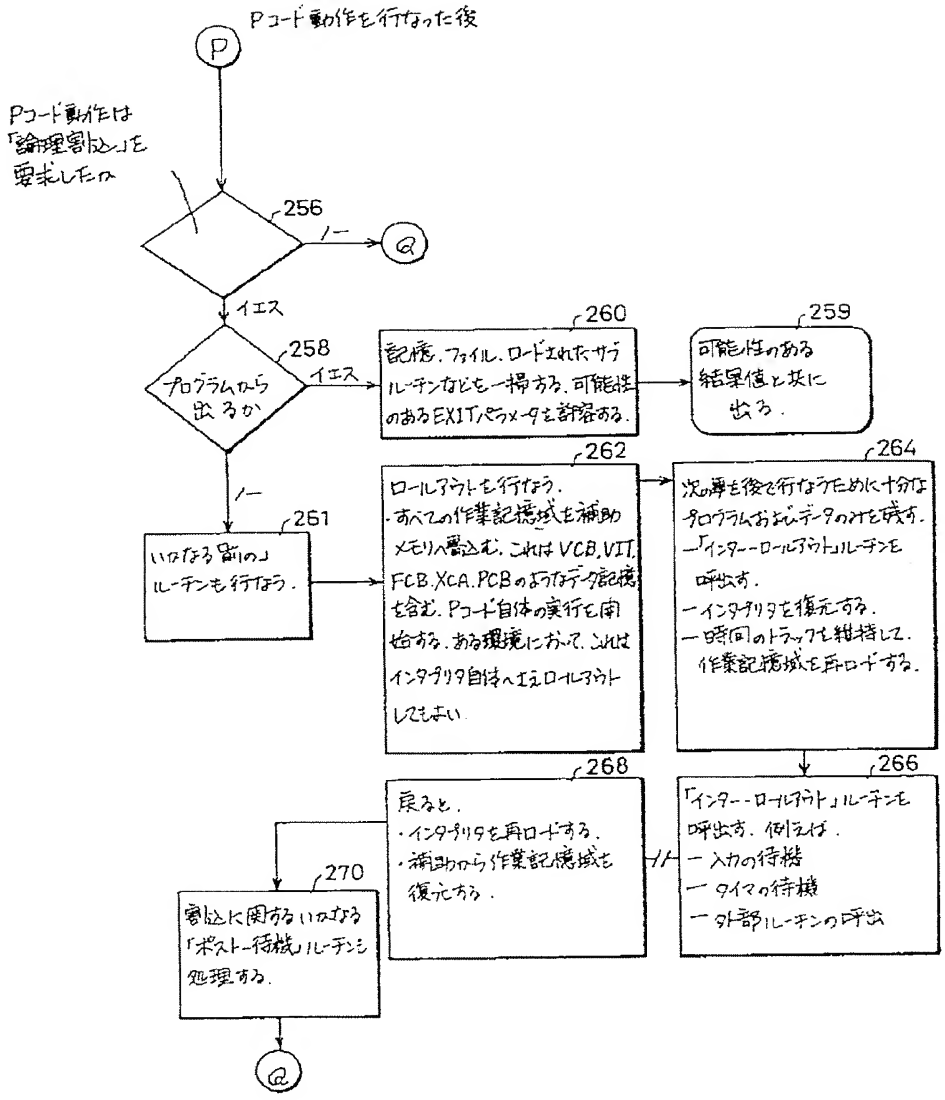
【図３４】



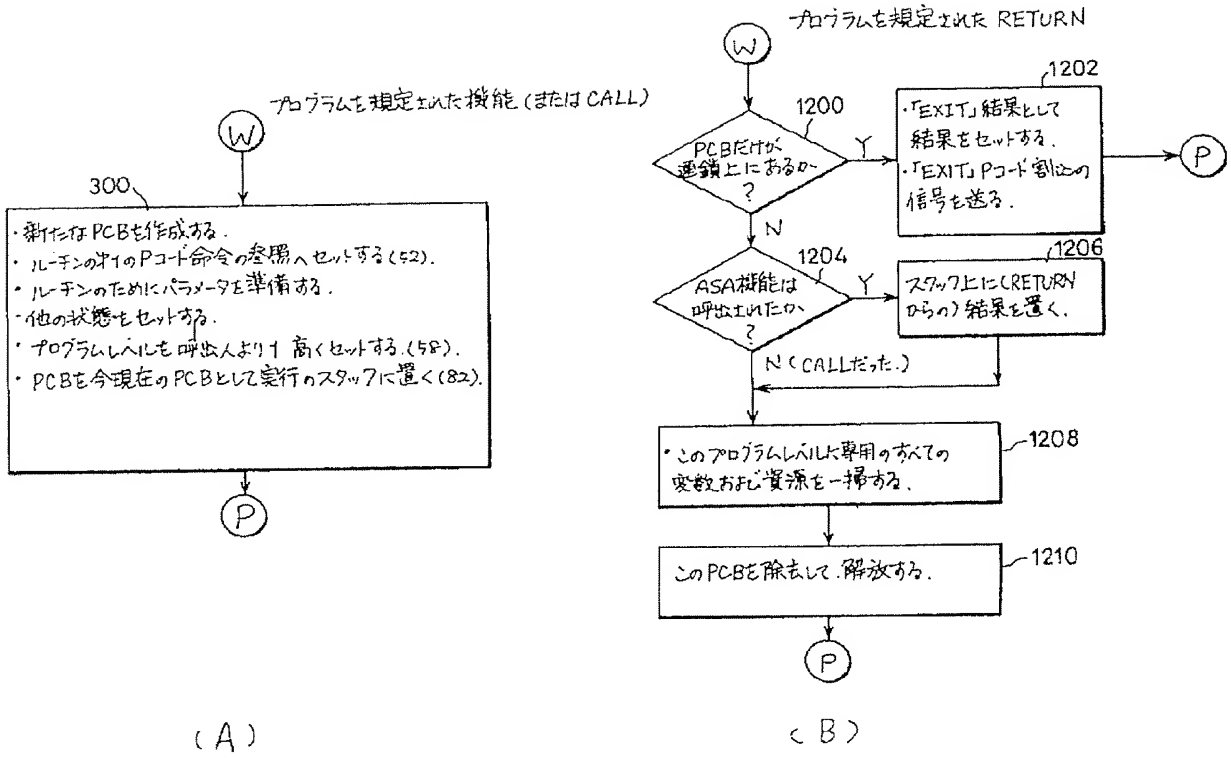
【図３５】



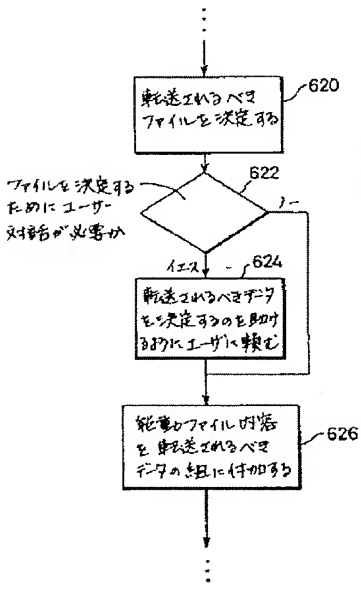
【図１５】


Pコード動作を行なった後
P
Pコード動作は「論理割込」を要求したか
256
ノー
Q
イエス
258
プログラムから出るか
イエス
260
記憶、ファイル、ロードされたサブルーチンなどを一掃する、可能性のあるEXITパラメータを許容する。
259
可能性のある結果値と共に出る。
ノー
261
いかなる留の」ルーチンも行なう。
262
ロールアウトを行なう。
・すべての作業記憶域を補助メモリへ書込む。これはVCB,VIT,FCB,XCA,PCBのようなデータ記憶を含む。Pコード自体の実行を開始する。ある環境において、これはインタプリタ自体へエロールアウトしてもよい
264
次の事を後で行なうために十分なプログラムおよびデータのみを残す。
—「インターロールアウト」ルーチンを呼出す。
—インタプリタを復元する。
—時間のトラックを維持して、作業記憶域を再ロードする。
266
「インターロールアウト」ルーチンを呼出す。例えば、
—入力の待機
—タイマの待機
—外部ルーチンの呼出
268
戻ると、
・インタプリタを再ロードする。
・補助から作業記憶域を復元する。
270
割込に関するいかなる「ポストー待機」ルーチンを処理する。
Q

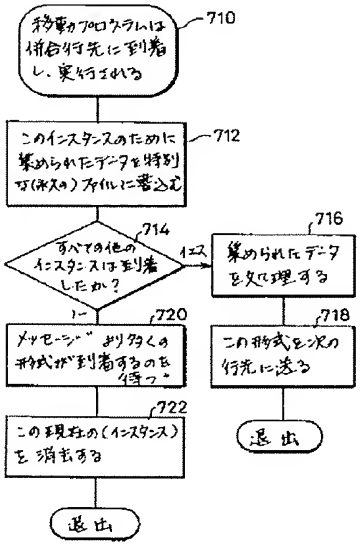
【図１６】


プログラムを規定された機能（またはCALL）
W
300
・新たなPCBを作成する．
・ルーチンのオイのPコード命令の番場へセットする(52)．
・ルーチンのためにパラメータを準備する．
・他の状態をセットする．
・プログラムレベルを呼出人より1 高くセットする．(58)．
・PCBを今現在のPCBとして実行のスタックに置く(82)．
P
プログラムを規定されたRETURN
W
1200
PCBだけが連鎖上にあるか？
Y
1202
・「EXIT」結果として結果をセットする．
・「EXIT」Pコード割込の信号を送る．
P
N
1204
ASA機能は呼出されたか？
Y
1206
スタック上に(RETURNからの)結果を置く．
N（CALLだった．）
1208
・このプログラムレベル専用のすべての変数および資源を一掃する．
1210
このPCBを除去して．解放する．
P
（A）
（B）

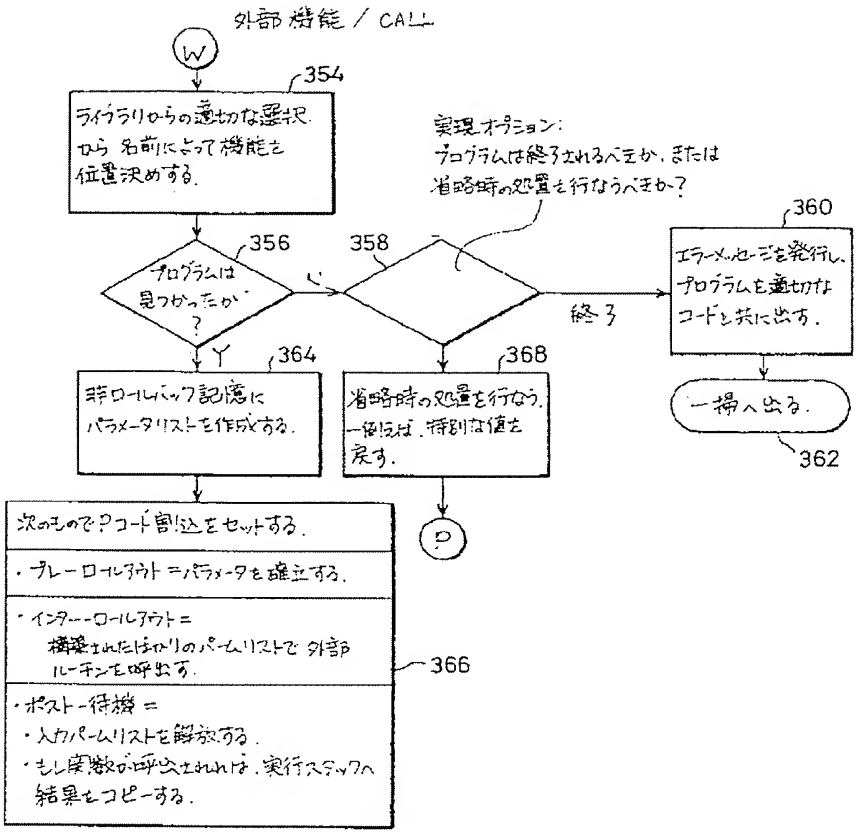
【図３３】


620
転送されるべきファイルを決定する
622
ファイルを決定するためにユーザー対話が必要か
イエス
624
転送されるべきデータを決定するのを助けるためにユーザに頼む
能動ファイル内容を転送されるべきデータの組に付加する
626

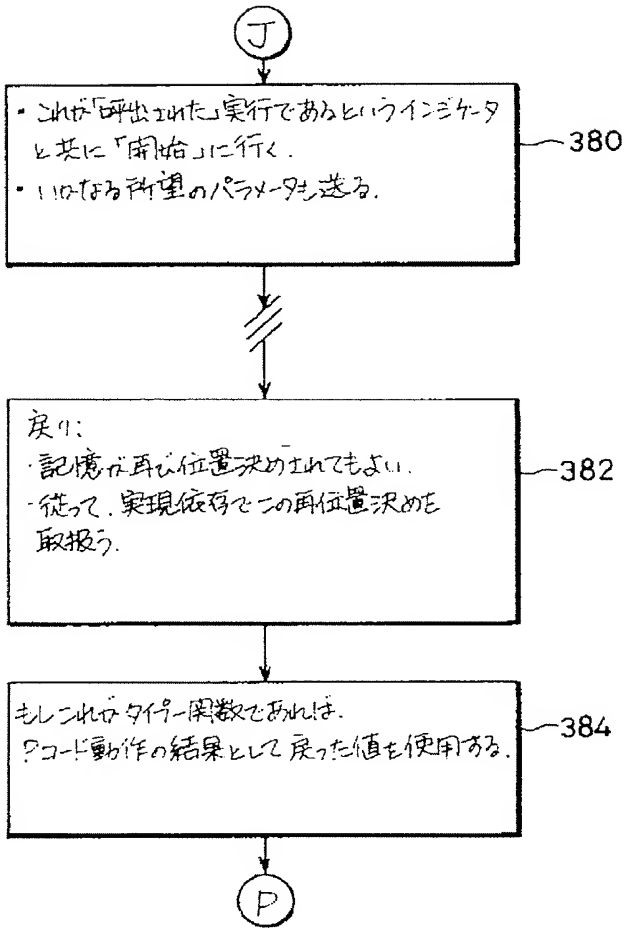
【図３８】


710
移動プログラムは併合行先に到着し、実行される
712
このインスタンスのために集められたデータを特別な(永久の)ファイルに書込む
714
すべての他のインスタンスは到着したか？
イエス
716
集められたデータを処理する
718
この形式をシンクの行先に送る
退出
720
メッセージより多くの形式が到着するのを待つ
722
この現在の(インスタンス)を消去する
退出

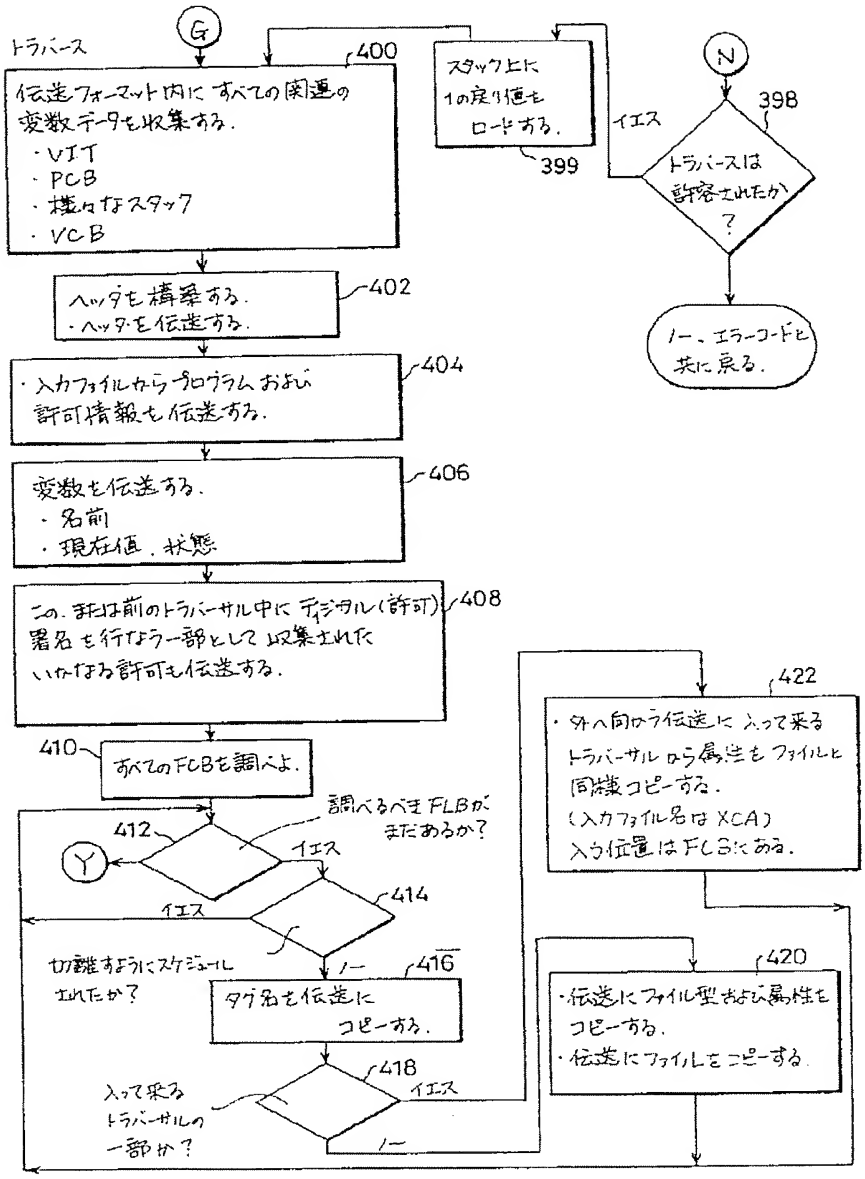
【図１８】


外部機能 / CALL
W
354
ライブラリからの適切な選択から 名前によって機能を位置決めする.
実現オプション:
プログラムは終了されるべきか、または省略時の処置を行なうべきか?
356
プログラムは見つかったか?
358
終了
360
エラーメッセージを発行し、プログラムを適切なコードと共に出す.
一掃へ出る.
362
Y
364
非ローカルバッファ記憶にパラメータリストを作成する.
368
省略時の処置を行なう. 一般には、特別な値を戻す.
P
次のもので?コード割込をセットする.
・プレーローカルアウト=パラメータを確立する.
・インターローカルアウト=
構築されたばかりのパームリストで外部ルーチンを呼出す.
366
・ポストー待機=
・入力パームリストを解放する.
・もし関数が呼出されれば、実行スタックへ結果をコピーする.

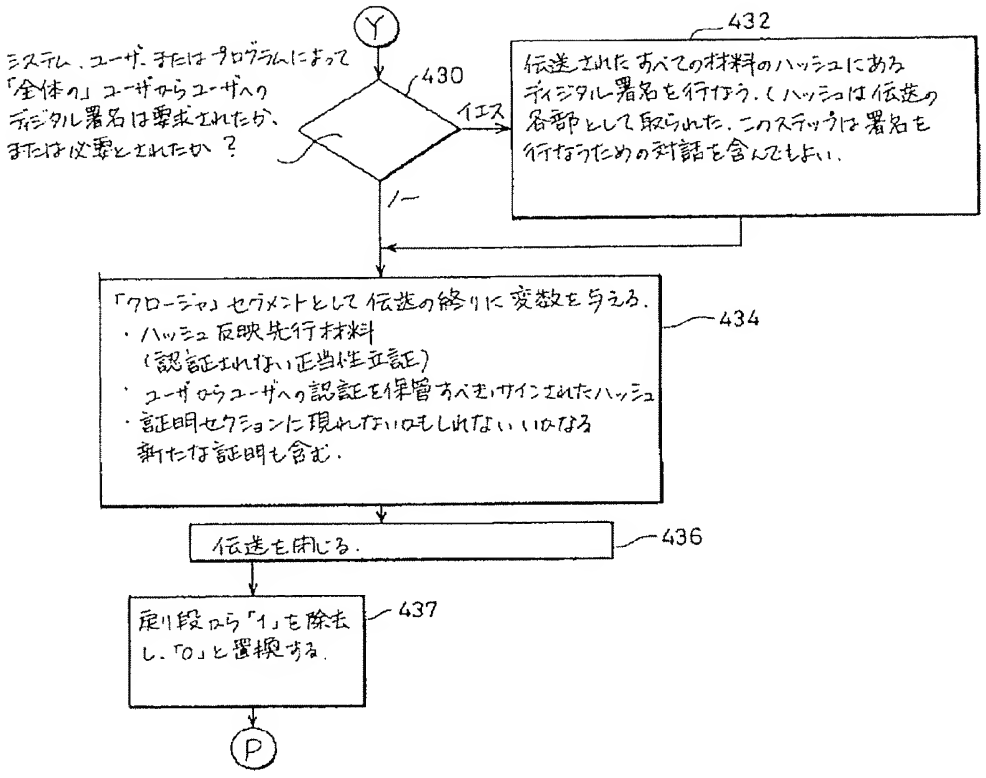
【図１９】


J
・これが「呼出された」実行であるというインジケータと共に「開始」に行く．
・いかなる所望のパラメータも送る．
380
戻り：
・記憶が再び位置決めされてもよい．
・従って，実現依存でこの再位置決めを取扱う．
382
もしこれがタイプ－関数であれば，
そコード動作の結果として戻った値を使用する．
384
P

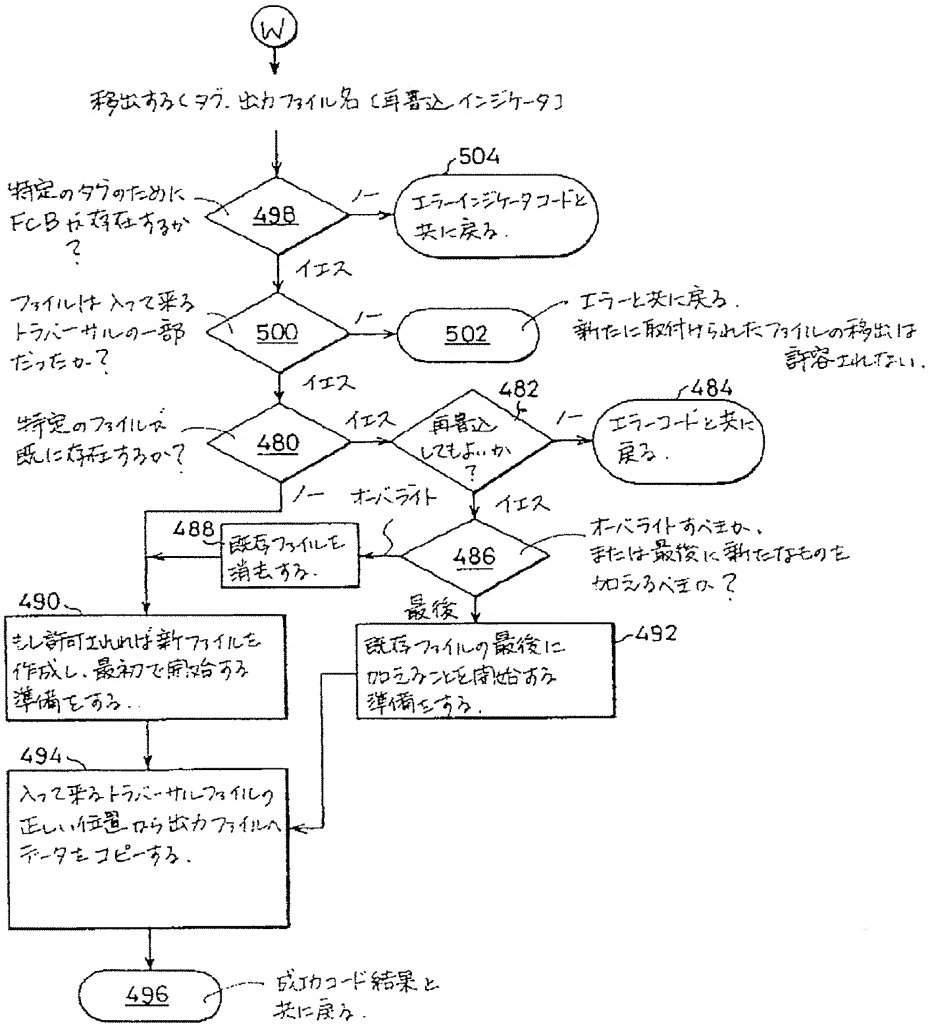
【図２０】


トラバース
G
400
伝送フォーマット内にすべての関連の変数データを収集する。
・VIT
・PCB
・様々なスタック
・VCB
402
ヘッダも構築する。
・ヘッダを伝送する。
404
・入力ファイルからプログラムおよび許可情報も伝送する。
406
変数を伝送する。
・名前
・現在値、状態
408
この、または前のトラバーサル中にデジタル(許可)署名を行なう一部として収集された いかなる許可も伝送する。
410
すべてのFCBを調べよ。
412
調べるべきFLBがまだあるか?
Y
イエス
414
イエス
切離すようにスケジュールされたか?
ノー
416
タグ名を伝送にコピーする。
418
入って来るトラバーサルの一部か?
イエス
ノー
422
・外へ向かう伝送に入って来るトラバーサルから属性をファイルと同様コピーする。
(入力ファイル名はXCA)
入り位置はFLBにある。
420
・伝送にファイル型および属性をコピーする。
・伝送にファイルをコピーする。
スタック上に1の戻り値をロードする。
399
イエス
N
398
トラバースは許容されたか?
ノー、エラーコードと共に戻る。

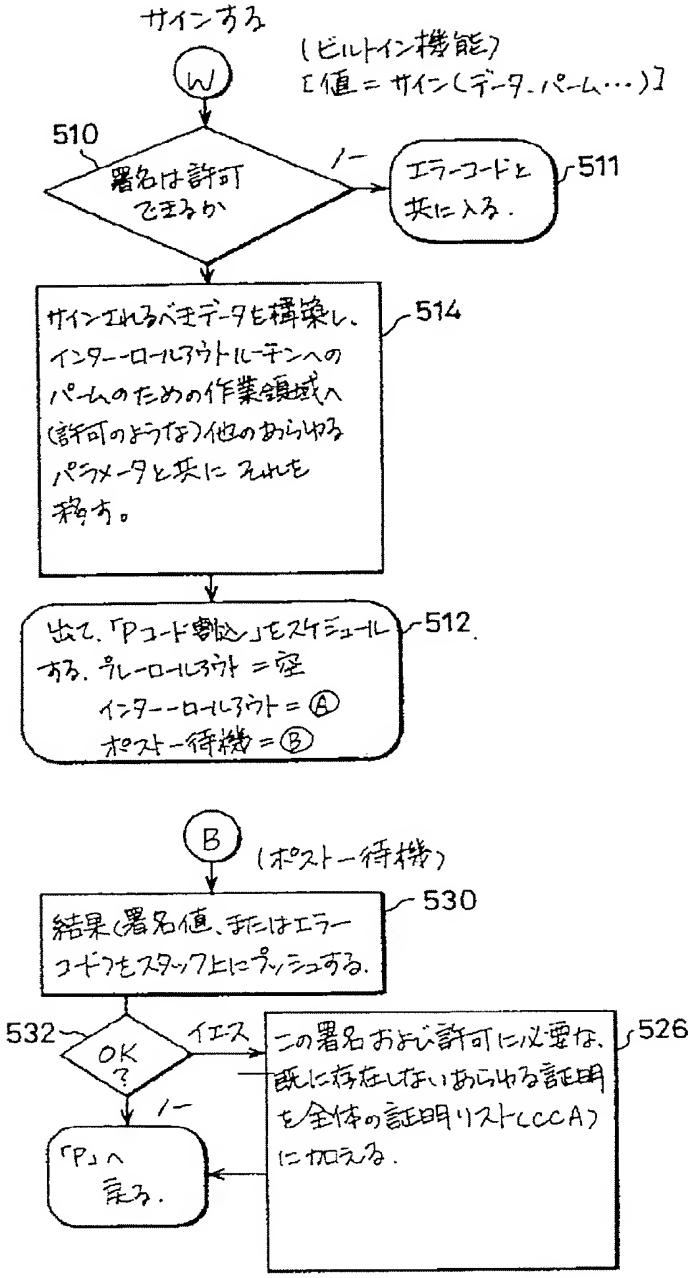
【図２１】


Y
システム、ユーザ、またはプログラムによって「全体の」ユーザからユーザへのディジタル署名は要求されたか、または必要とされたか？
430
イエス
ノー
432
伝送されたすべての材料のハッシュにあるディジタル署名を行なう。(ハッシュは伝送の各部として取られた。このステップは署名を行なうための対話を含んでもよい。
「クローザ」セグメントとして伝送の終りに変数を与える。
・ハッシュ反映先行材料
(認証されない正当性立証)
・ユーザからユーザへの認証を保管すべきサインされたハッシュ
・証明セクションに現れないかもしれないいかなる新たな証明も含む。
434
伝送を閉じる。
436
底り段から「1」を除去し、「0」と置換する。
437
P

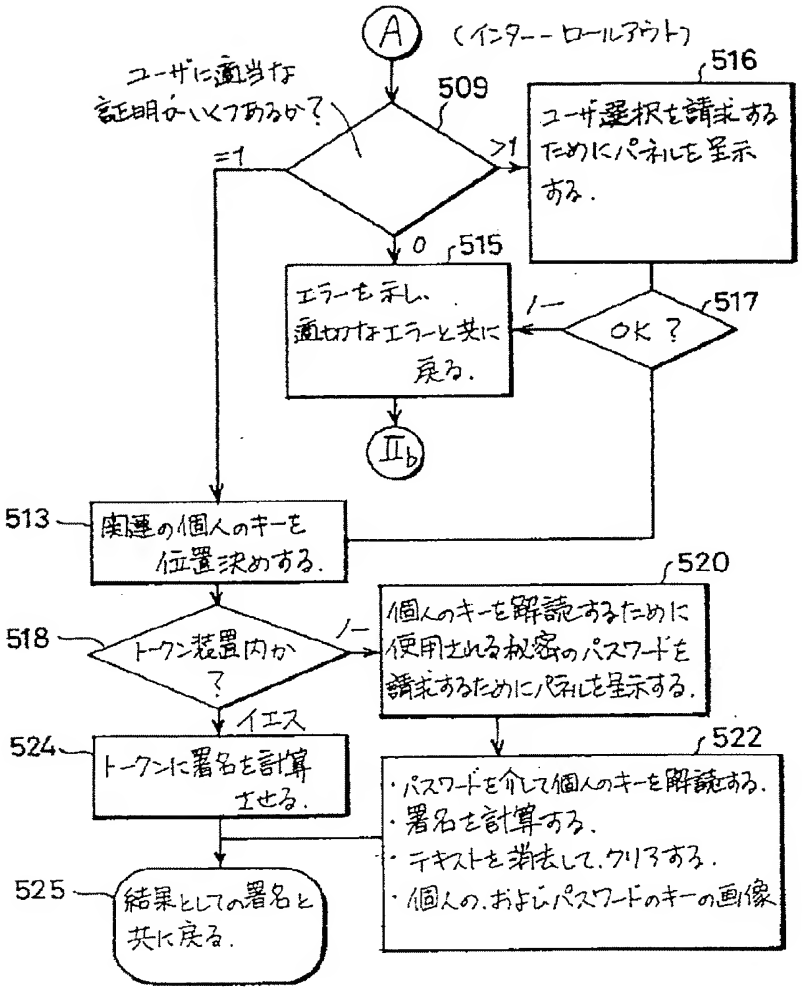
【図２５】


W
移出するタブ、出力ファイル名〔再書込インジケータ〕
特定のタブのためにFCBが存在するか？
498
ノー
504
エラーインジケータコードと共に戻る。
イエス
ファイルは入って来るトラバーサルの一部だったか？
500
ノー
502
エラーと共に戻る。新たに取付けられたファイルの移出は許容されない。
イエス
特定のファイルが既に存在するか？
480
イエス
再書込してもよいか？
482
ノー
484
エラーコードと共に戻る。
ノー
オーバライト
イエス
488
既存ファイルを消去する。
486
オーバライトすべきか、または最後に新たなものを加えるべきか？
490
もし許可されれば新ファイルを作成し、最初で開始する準備をする。
最後
492
既存ファイルの最後に加えることを開始する準備をする。
494
入って来るトラバーサルファイルの正しい位置から出力ファイルへデータをコピーする。
496
成功コード結果と共に戻る。

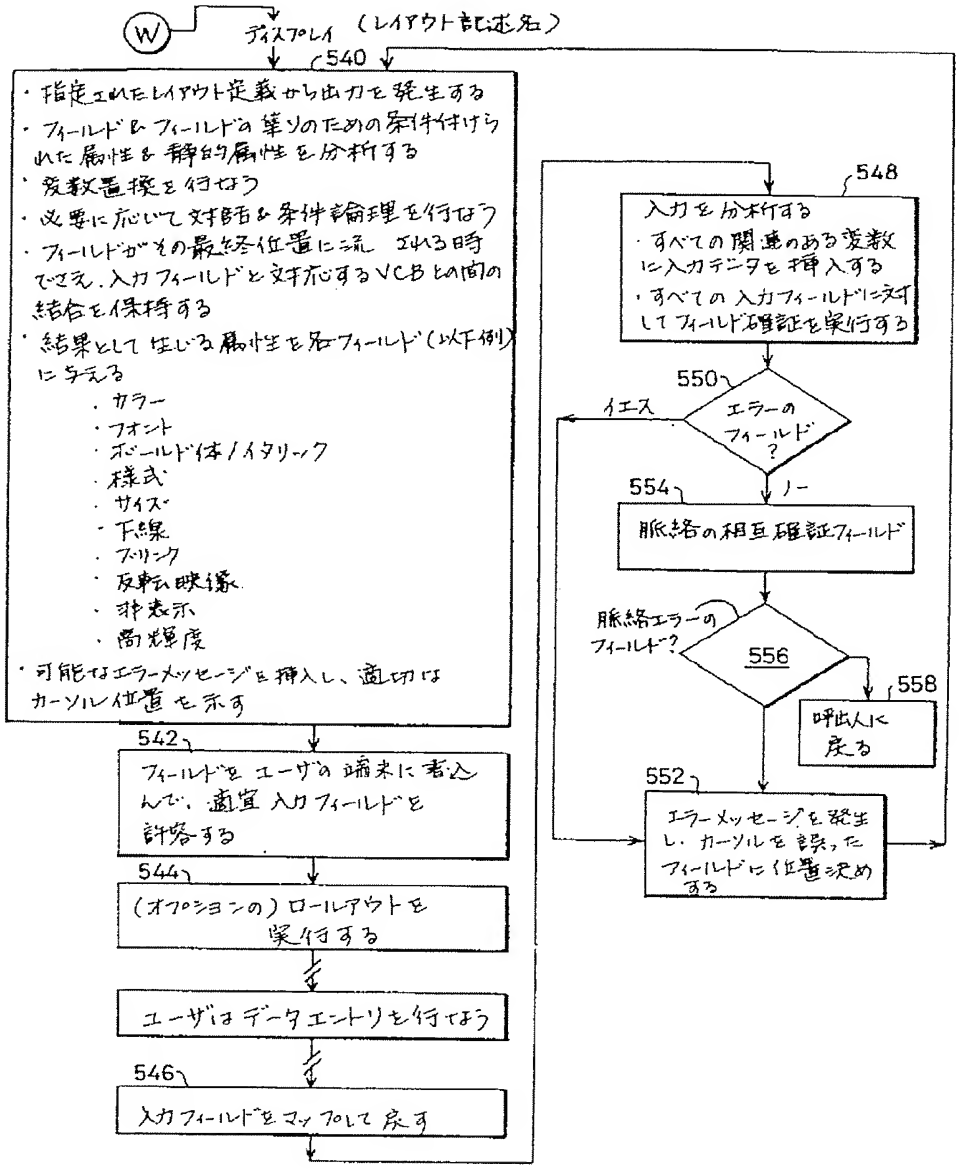
【図２６】


サインする
(ビルトイン機能)
[値＝サイン(データ、パーム…)]
W
510
署名は許可できるか
ノー
エラーコードと共に入る.
511
サインされるべきデータを構築し、インターロールアウトルーチンへのパームのための作業領域へ(許可のような)他のあらゆるパラメータと共にそれを格す。
514
出て、「Pコード割込」をスケジュールする. プレーロールアウト＝空
インターロールアウト＝Ⓐ
ポストー待機＝Ⓑ
512
B
(ポストー待機)
結果(署名値、またはエラーコードをスタック上にプッシュする.
530
532
OK?
イエス
この署名および許可に必要な、既に存在しないあらゆる証明を全体の証明リスト(CCA)に加える.
526
ノー
「P」へ戻る.

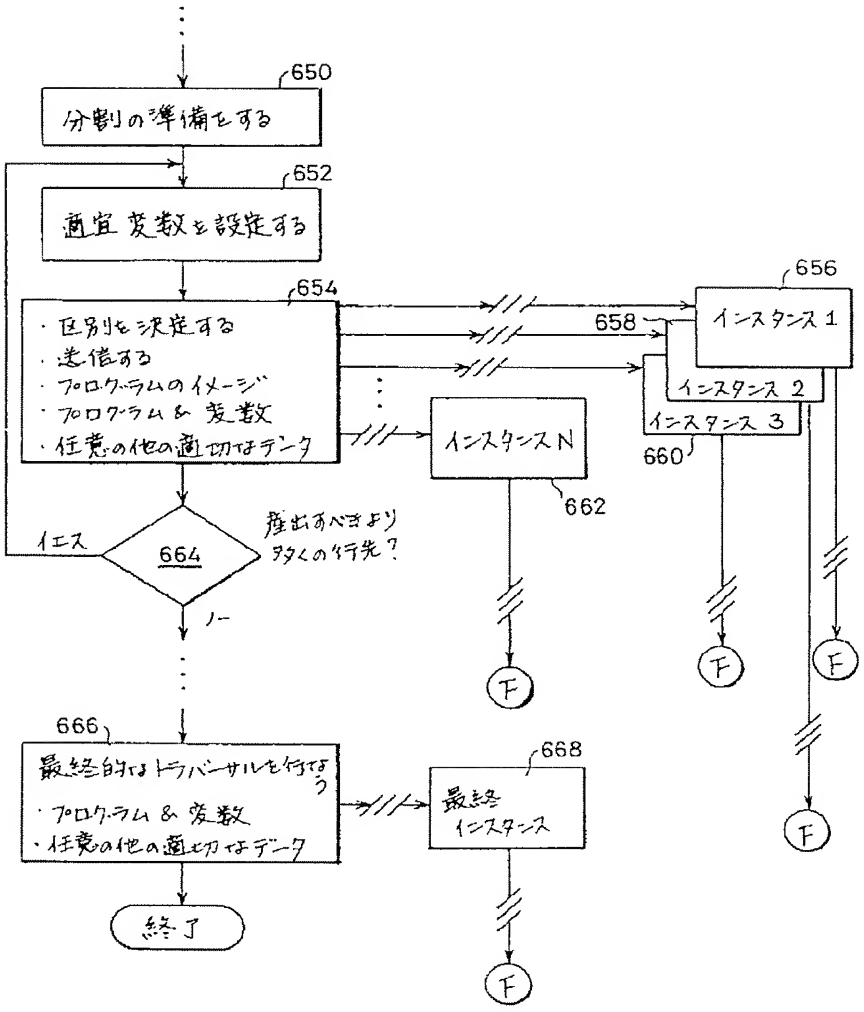
【図２７】


A
(インターロールアウト)
ユーザに適当な証明がいくつあるか?
509
=1
>1
0
516
ユーザ選択を請求するためにパネルを呈示する.
515
エラーを示し、適切なエラーと共に戻る.
ノー
517
OK?
IIb
513
関連の個人のキーを位置決めする.
520
個人のキーを解読するために使用される秘密のパスワードを請求するためにパネルを呈示する.
518
トークン装置内か?
ノー
イエス
524
トークンに署名を計算させる.
522
・パスワードを介して個人のキーを解読する.
・署名を計算する.
・テキストを消去して、クリアする.
・個人の、およびパスワードのキーの画像
525
結果としての署名と共に戻る.

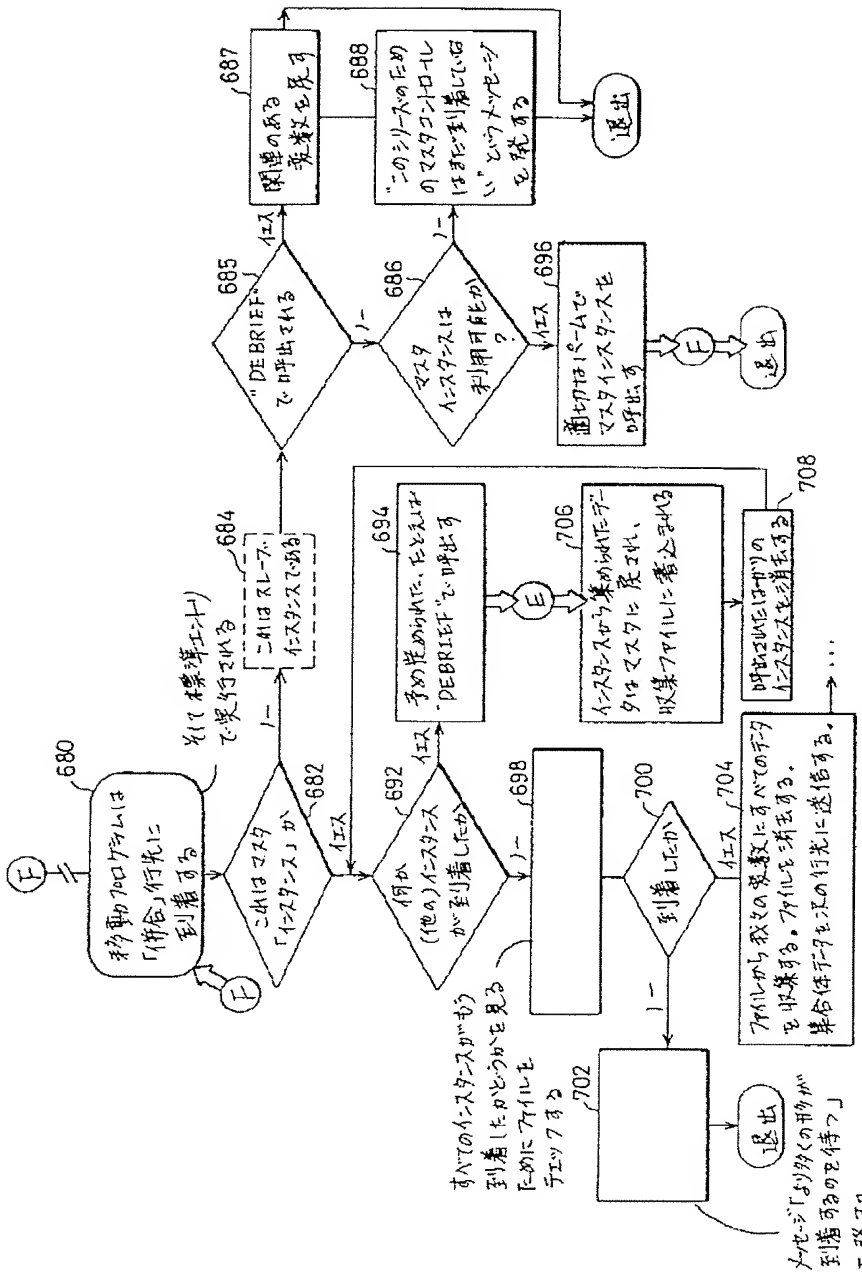
【図２８】


W
ディスプレイ（レイアウト記述名）
540
・指定された レイアウト定義から出力を発生する
・フィールド & フィールドの集りのための条件付けられた属性 & 静的属性を分析する
・変数置換を行なう
・必要に応じて対話 & 条件論理を行なう
・フィールドがその最終位置に流れる時でさえ、入力フィールドと対応するVCBとの間の結合を保持する
・結果として生じる属性を各フィールド（以下例）に与える
・カラー
・フォント
・ボールド体/イタリック
・様式
・サイズ
・下線
・ブリンク
・反転映像
・非表示
・高輝度
・可能なエラーメッセージを挿入し、適切ならカーソル位置を示す
542
フィールドをユーザの端末に書込んで、適宜入力フィールドを許容する
544
（オプションの）ロールアウトを実行する
ユーザはデータエントリを行なう
546
入力フィールドをマップして戻す
548
入力を分析する
・すべての関連のある変数に入力データを挿入する
・すべての入力フィールドに対してフィールド確証を実行する
550
エラーのフィールド？
イエス
ノー
554
脈絡の相互確証フィールド
脈絡エラーのフィールド？
556
558
呼出人に戻る
552
エラーメッセージを発生し、カーソルを誤ったフィールドに位置決めする

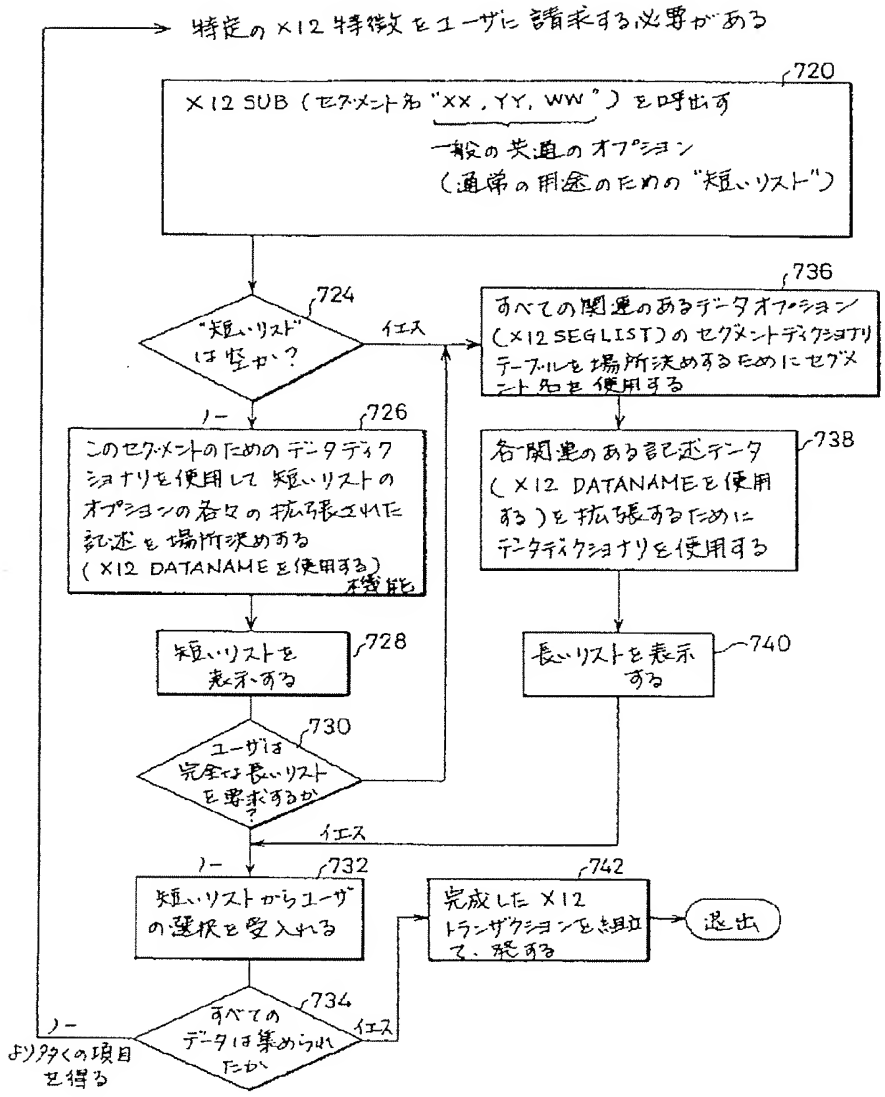
【図３６】


650
分割の準備をする
652
適宜 変数を設定する
654
・区別を決定する
・送信する
・プログラムのイメージ
・プログラム & 変数
・任意の他の適切なデータ
656
インスタンス 1
658
インスタンス 2
インスタンス 3
660
インスタンスN
662
イエス
664
産出すべきより多くの行先？
ノー
666
最終的なトラバーサルを行なう
・プログラム & 変数
・任意の他の適切なデータ
668
最終インスタンス
終了
F

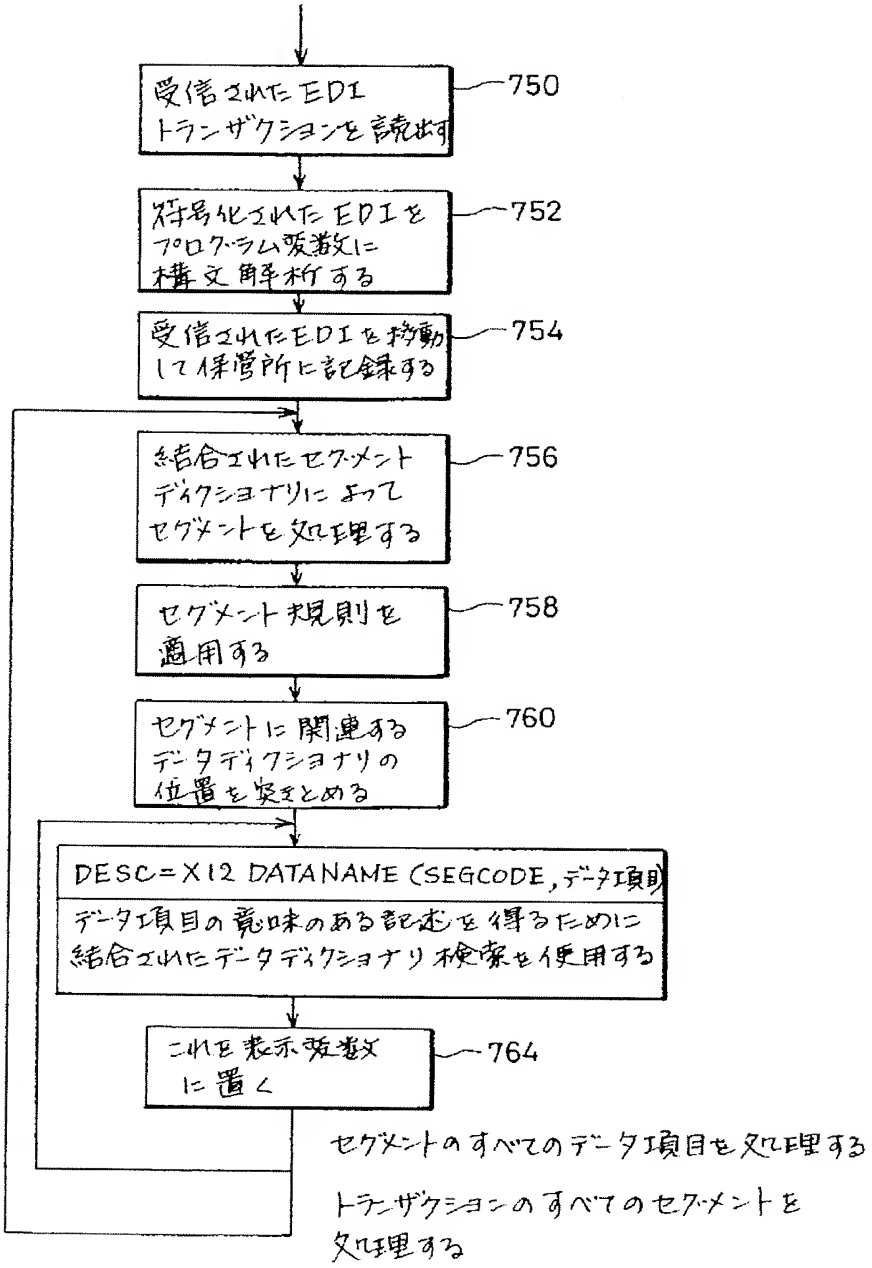
【図３７】


680
E
F
そして標準エントリ
で突合される
684
これはスレーブ
インスタンスである
682
これはマスタ
インスタンスか
イエス
ノー
692
何か
(他の)インスタンス
が到着したか
イエス
ノー
694
予め定められた、たとえば
"DEBRIEF"を呼出す
E
706
インスタンスから集められたデータは
マスタに戻され、
収集ファイルに書込まれる
708
呼出されたばかりの
インスタンスを消去する
698
700
到着したか
イエス
ノー
704
ファイルから残りの変数にすべてのデータを収集する。ファイルを消去する。
集合体データをその行先に送信する。
...
すべてのインスタンスがもう
到着したかどうかを見る
ためにファイルを
チェックする
702
退出
メッセージ「しばらくの間が
到着するのを待つ」
を表示する
685
"DEBRIEF"
を呼出されるか
イエス
ノー
687
関連のある
変数を戻す
688
"このシリーズのための
マスタコントロールは
まだ到着していない"
というメッセージ
を表示する
退出
686
マスタ
インスタンスは
利用可能か
?
イエス
ノー
696
適切なパームで
マスタインスタンスを
呼出す
F
退出

【図３９】


特定のX12特徴をユーザに請求する必要がある
720
X12 SUB（セグメント名 "XX, YY, WW"）を呼出す
一般の共通のオプション
（通常の用途のための"短いリスト"）
724
"短いリスト"は空か？
イエス
ノー
736
すべての関連のあるデータオプション（X12 SEGLIST）のセグメントディクショナリテーブルを場所決めするためにセグメント名を使用する
726
このセグメントのためのデータディクショナリを使用して短いリストのオプションの各々の拡張された記述を場所決めする（X12 DATANAMEを使用する）
738
各関連のある記述データ（X12 DATANAMEを使用する）を拡張するためにデータディクショナリを使用する
728
短いリストを表示する
740
長いリストを表示する
730
ユーザは完全な長いリストを要求するか？
イエス
ノー
732
短いリストからユーザの選択を受入れる
742
完成したX12トランザクションを組立て、発する
退出
734
すべてのデータは集められたか
イエス
ノー
より多くの項目を得る

【図４０】



フロントページの続き

(51) Int.Cl.[5]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｈ０４Ｌ 9/12				